

# American National Standard

for information systems —

information resource dictionary system  
(IRDS)

ANSI X3.138-1988



american national standards institute, inc.  
1430 broadway, new york, new york 10018





**American National Standard  
for Information Systems —  
Information Resource Dictionary System  
(IRDS)**

Secretariat

**Computer and Business Equipment Manufacturers Association**

Approved October 19, 1988

**American National Standards Institute, Inc**

**Abstract**

This standard specifies a Command Language Interface and an (interactive) Panel Interface to the Information Resource Dictionary System (IRDS), a software tool which can be used to control, describe, protect, document, and facilitate use of an installation's information resources. The standard is composed of seven Modules: a core standard (Module 1), and six optional Modules which extend the facilities available in the core standard. The optional Modules are dependent on the availability of the core standard.

# American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

**CAUTION NOTICE:** This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Published by

**American National Standards Institute  
1430 Broadway, New York, New York 10018**

Copyright © 1989 by American National Standards Institute, Inc  
All rights reserved.

No part of this publication may be reproduced in any form,  
in an electronic retrieval system or otherwise, without  
the prior written permission of the publisher.

Printed in the United States of America

PC1½M189/65

# Foreword

(This Foreword is not part of American National Standard X3.138-1988.)

This document is entitled American National Standard for Information Systems - Information Resource Dictionary System (IRDS). It contains the specification of a computer software system which provides the facilities for capturing, modifying, managing, and disseminating the specifications of information and information processing resources.

This standard contains eight chapters with the following titles:

- Requirements for a Conformant Implementation
- Module 1 - Core Standard
- Module 2 - Basic Functional Schema
- Module 3 - IRDS Security
- Module 4 - Extensible Life Cycle Phase Facility
- Module 5 - IRDS Procedures
- Module 6 - Application Program Interface
- Module 7 - Entity Lists

The chapter entitled "Requirements for a Conformant Implementation" specifies these requirements for all of the Modules of the standard. All conformant implementations shall supply Module 1, the core standard. The remaining six Modules are optional subject to the interrelationships and dependencies specified.

In developing the specifications for a standard IRDS, it was recognized that IRDS technology is still evolving and that uses for IRDS technology are continually expanding. Thus, it was necessary to provide a mechanism that allowed the establishment of a standard that would evolve as IRDS technology evolves. In addition, it was recognized that any one installation wanting to purchase an IRDS might not need all of the functionality that could be provided by an IRDS; this is particularly apparent when one examines the potential implementation environments for an IRDS, including the microcomputer environment. This approach allows the optional Modules to be obtained separately as users encounter the need for additional functionality.

Information contained in an Appendix in these specifications is not considered part of this standard but is rather auxiliary to the standard.

Future control of this document will reside with Accredited Standards Committee X3, Information Processing Systems.

Suggestions for improvement of this standard will be welcome. They should be sent to: X3 Secretariat/CBEMA, Suite 500, 311 First Street NW, Washington, DC 20001.

This standard was processed and approved for submittal to ANSI by the Accredited Standards Committee X3, Information Processing Systems. Committee approval of this standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

Richard Gibson, Chair  
Donald C. Loughry, Vice-Chair  
Catherine A. Kachurik, Administrative Secretary

<u>Organizations Represented</u>	<u>Name of Representative</u>
American Library Association	Paul Peters
American Nuclear Society	Geraldine C. Main
	Sally Hartzell (Alt)
AMP Incorporated	Edward Kelly
	Ronald Lloyd (Alt)
Apple	Karl Kimball
	Michael J. Lawler (Alt)
Association of the Institute for Certification of Computer Professionals	Thomas M. Kurihara
AT&T	Paul D. Bartoli
	Thomas F. Frost (Alt)
Control Data Corporation	Charles E. Cooper
	Keith Lucke (Alt)
Cooperating Users of Burroughs Equipment	Thomas Easterday
	Donald Miller (Alt)
Data General Corporation	Lee Schiller
	Lyman Chapin (Alt)
Digital Equipment Computer Users Society	Dennis Perry
Digital Equipment Corporation	Gary S. Robinson
	Delbert L. Shoemaker (Alt)
Eastman Kodak	Gary Haines
	Charleton C. Bard (Alt)
GUIDE International	Frank Kirshenbaum
	Sandra Swartz Abraham (Alt)
Hewlett-Packard	Donald C. Loughry
Honeywell Bull	David M. Taylor
IBM Corporation	Mary Anne Gray
	Robert H. Follett (Alt)
IEEE Computer Society	Sava I. Sherr
	Thomas A. Varetoni (Alt)
	Helen Wood (Alt)
Lawrence Berkeley Laboratory	David F. Stevens
	Robert L. Fink (Alt)
MAP/TOP	James D. Converse
	Mike Kaminski (Alt)
Moore Business Forms	Delmer H. Oddy
National Communications System	Dennis Bodson
	George White (Alt)
National Institute of Standards and Technology	Robert E. Rountree
	James H. Burrows (Alt)
NCR Corporation	William E. Snyder
	A. R. Daniels (Alt)
OMNICOM	Harold C. Folts
	Catherine Howells (Alt)
Prime Computer, Inc.	Arthur Norton
	Donna A. Poulack (Alt)

Scientific Computer Systems Corporation

SHARE, Inc.

3M Company

Travelers Insurance Companies, Inc.

Unisys

U.S. Department of Defense

U.S. General Services Administration

VIM

Wang Corporation

Xerox Corporation

James A. Baker

Carl Haberland (Alt)

Thomas B. Steel

Robert A. Rannie (Alt)

Paul D. Jahnke

Joseph T. Brophy

Marvin W. Bass

Stanley Fenner (Alt)

Fred Virtue

Belkis Leong-Hong (Alt)

William C. Rinehuls

Larry L. Jackson (Alt)

Chris Tanner

Madeleine Sparks (Alt)

J.J. Cinecoe

Sarah Wagner (Alt)

John L. Wheeler

Roy Pierce (Alt)

At the time it approved this standard, the Technical Committee X3H4 on Information Resource Dictionary System had the following members:

Anthony J. Winkler, Chair

David T. Carpenter, Vice-Chair

Donald J. McCaffrey, Secretary

Hugh R. Beyer

William Chapman

Ewart E. Cutway, Jr.

Alan H. Goldfine

Fred Green

Ray J. Hollenback

Henry C. Lefkovits

John L. McCarthy

Joan Monia

Dan Riehle

Daniel Teichroew

David Thomas

E. James Emerson (Alt)

George E. Gajnak (Alt)

Fredric C. Gey (Alt)

Judith J. Newton (Alt)

Jim O'Leary (Alt)

Richard A. Peik (Alt)

Hasan H. Sayani (Alt)

American National Standard  
for Information Systems —

Information Resource Dictionary System  
(IRDS)

**Table of Contents**  
**American National Standard X3.138-1988**

<b>CHAPTER</b>		<b>PAGE</b>
	Requirements for a Conformant Implementation . . . . .	1
1	Module 1 - Core Standard . . . . .	1-1
2	Module 2 - Basic Functional IRD Schema . . . . .	2-1
3	Module 3 - IRDS Security . . . . .	3-1
4	Module 4 - Extensible Life Cycle Phase Facility . . . . .	4-1
5	Module 5 - IRDS Procedures . . . . .	5-1
6	Module 6 - Applications Program Interface . . . . .	6-1
7	Module 7 - Entity Lists . . . . .	7-1

(NOTE: A detailed table of contents is provided at the beginning of each Chapter.)

# Requirements for a Conformant Implementation

## CONTENTS

SECTION	PAGE
1 Scope and Field of Application of this American National Standard . . . . .	1
2 Background . . . . .	1
3 The Architecture of the IRDS . . . . .	1
3.1 The Entity-Relationship Data Model . . . . .	3
3.2 The Four-Level Architecture . . . . .	3
3.3 The Information Resource Dictionary Schema . . . . .	4
3.4 Extending the IRD Schema . . . . .	6
3.5 The Information Resource Dictionary . . . . .	7
3.6 The Information Resource Dictionary System . . . . .	7
4 Definition of a Conformant Implementation of the IRDS . . . . .	8
4.1 Definition of the Standard IRDS . . . . .	8
4.2 Definition of a Conformant Implementation . . . . .	10
4.2.1 Standard/Extended Mode . . . . .	10
4.2.2 Substitute or Additional Language Elements in the Command Language Interface . . . . .	11
4.2.3 Character Substitution in the Command Language . . . . .	11
4.3 Relationship of a Conformant Program to a Conformant Implementation . . . . .	11
4.4 Relationship of a Conformant IRDS Implementation to a System Environment . . . . .	11
4.5 Scope of the Panel Interface . . . . .	12
Figures	
Figure 1 - The Four-Level Architecture of the IRDS Database . . . . .	2
Figure 2 - Associations of IRD Schema Entities . . . . .	5





## **Requirements for a Conformant Implementation**

### **1 Scope and Field of Application of this American National Standard**

This standard establishes the requirements for an Information Resource Dictionary System (IRDS), a software tool which can be used to control, describe, protect, document, and facilitate use of an installation's information resources. This standard assumes no implementation environment and assumes no run-time or compile-time dependent interfaces.

### **2 Background**

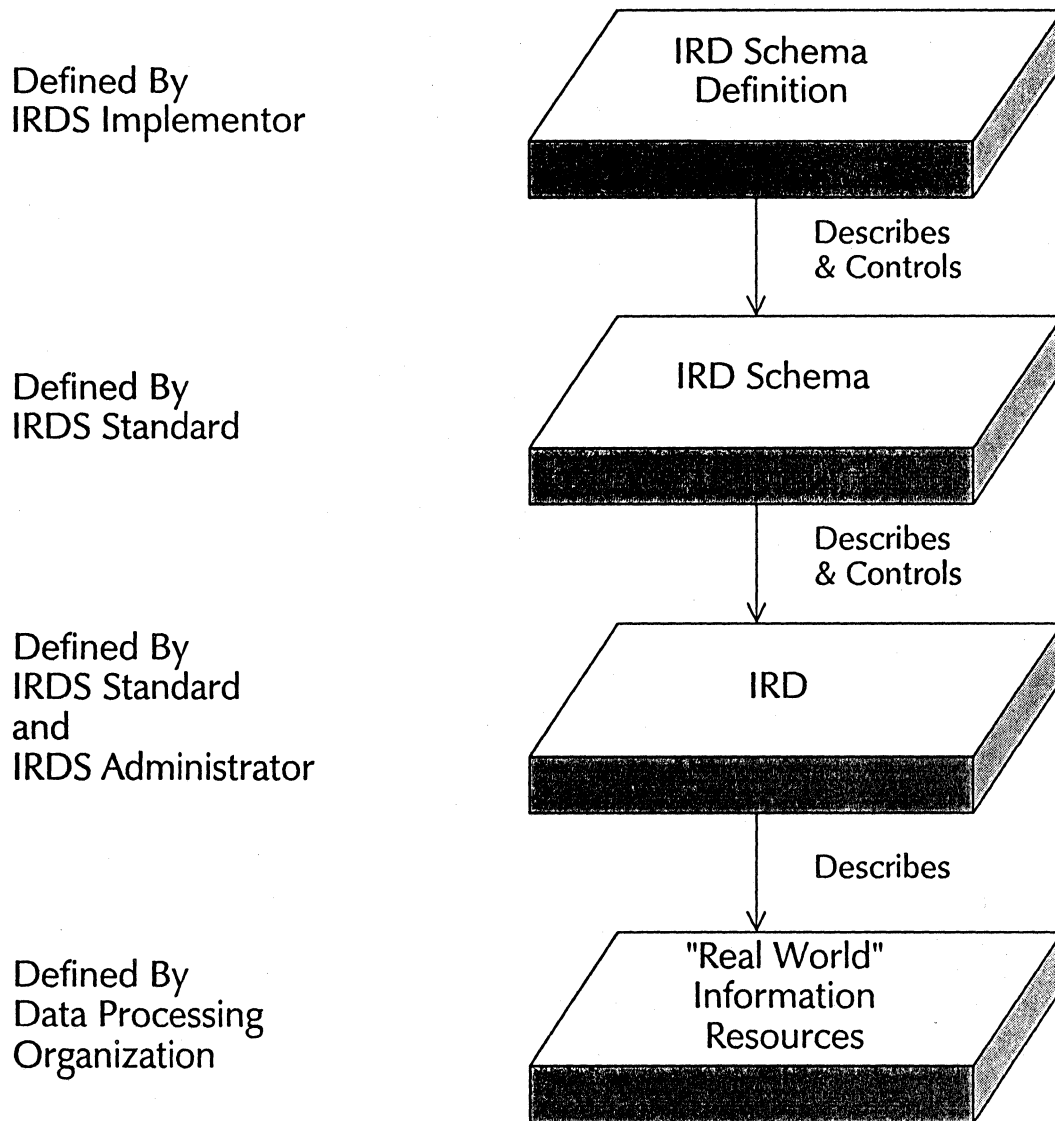
Early in the development of the IRDS specification, it was recognized that the IRDS was many things to many people. For example, identified applications of an IRDS include usage as or in support of:

- (1) A documentation tool.
- (2) A software life cycle and project management tool.
- (3) A data element standardization and management system.
- (4) An organizational planning tool.
- (5) A tool to support database administration, document administration, information resource management and data administration.
- (6) A tool for supporting a distributed processing and database environment.
- (7) A source and object library management system.
- (8) A configuration management facility.

It was also recognized that the developing body for this standard did not have the resources and expertise necessary to solve and specify solutions for all of the areas noted above. The primary objective, therefore, was to develop a specification which could serve as a foundation for satisfying recognized requirements for IRDSs, and which was sufficiently adaptable to accommodate future requirements. There was no intent to address the myriad of possible environments, such as those identified above, because the standard might never be finished, and in that event the problems would be attacked without availability of a core facility to serve as a foundation for extended functionality.

### **3 The Architecture of the IRDS**

The IRDS Database can be viewed as a four-level architecture in which the information specified at one level describes (and potentially controls) the information stored at the next lower level. Thus, one level defines the types of "objects" which can be described at the next lower level, and that level contains the "instances" of those types. These four levels are illustrated in Figure 1.



**Figure 1**  
**The Four-Level Architecture of the IRDS Database**

### 3.1 The Entity-Relationship Data Model

This standard uses an Entity-Relationship (E-R) data modeling approach to describe the contents of both the Information Resource Dictionary (IRD) and the IRD Schema, which defines the structure of the IRD. It is strongly typed, i.e., every object has one and only one type.

There is no intent to imply the actual implementation of an E-R database system. This should be evident, since the IRDS is foremost an application system which may be implemented using a database language and/or programming language with file manipulation facilities; it is not a database system.

The E-R data model used in this document is based on binary relationships between entities, and it allows attributes for both entities and relationships. The reason for choosing an E-R data model as the means for describing the IRD and its structure is that the majority of current implementations of IRDSs either use or are readily described with this approach. In addition, the developers felt that since there was no standard for specifying the data model of a standard, the best approach would be to use one which was:

- (1) Most natural for the problem being solved; i.e., in this case, the problem of managing objects, the associations among the objects, and their properties.
- (2) Easiest to understand for the end user of the IRDS.
- (3) More powerful and semantically richer than approaches used to model actual data; i.e., the relational and network data modeling approaches, since the approach chosen for the IRDS would have to be powerful enough to model the processes associated with the data, as well as the data modeling approach itself.

The binary relationship approach used in this specification was chosen because the vast majority of current implementations are based on binary relationships, and it is believed that the standard should be as simple as possible to use while providing the functionality necessary to satisfy the requirements of the "small" user.

### 3.2 The Four-Level Architecture

As shown in Figure 1, in this standard,

- (1) The top level of the four-level architecture is defined by the IRDS implementor. This level contains the types of all the objects which can be defined at the next lower level, the IRD Schema, the types of the relationships which can exist between these types, and certain properties of both of these types. These types are referred to as "meta-entity-types", "meta-relationship-types" and "meta-attribute-types", and are discussed in detail in Section 9 of Chapter 1. Additional types required for the Extensible Life Cycle Facility are discussed in Section 9 of Chapter 4. These types would be contained in the top level, if it were specified. It is not specified, because it would add unnecessary complexity to this standard.
- (2) The second level from the top is referred to as the Information Resource Dictionary (IRD) Schema. This level defines the types to be instantiated in the IRD. It also defines various control mechanisms, including naming rules, defaults, and validation information for the IRD contents.

- (3) The third level is the IRD. This level describes the environment being modeled. It describes the objects in the environment and the associations among those objects; the object descriptions are called entities, and the association descriptions are called relationships. This level also describes the properties of the objects and their associations; these properties are called attributes.
- (4) The fourth level, which is not described in the current standard, is the information resource environment; i.e., the "real world information resources", descriptions of which exist in the IRD.

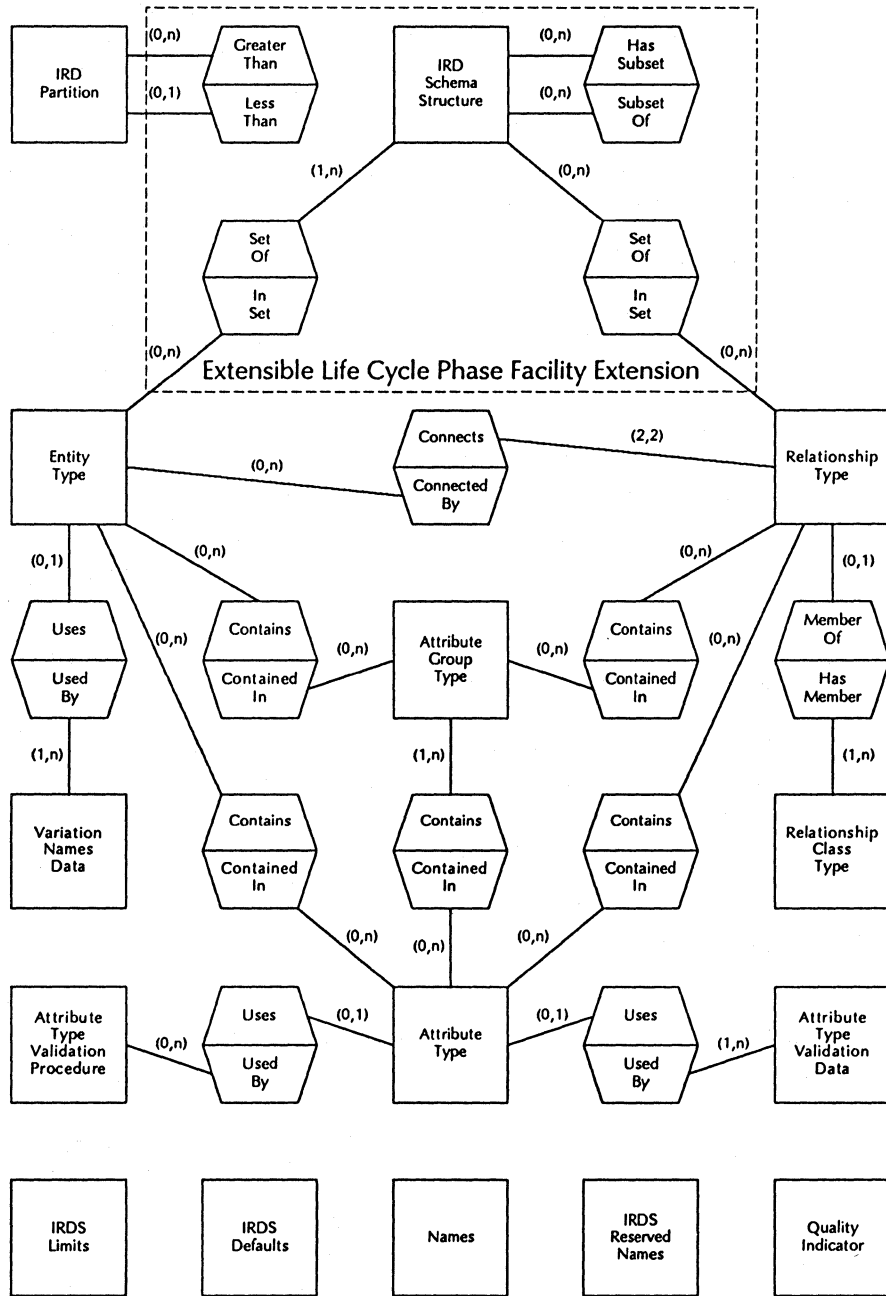
### 3.3 The Information Resource Dictionary Schema

The IRD Schema, the second level of the IRDS architecture, describes the structure and contents of the IRD. Thus, for every entity, relationship, attribute, and attribute-group that occurs in the IRD, the IRD Schema will contain a description of the corresponding entity-types, relationship-types, attribute-types, and attribute-group-types. The IRD Schema for Module 1 of this IRDS standard is the Minimal IRD Schema, which consists of those "meta-entities", "meta-relationships", and "meta-attributes" necessary to establish controls over the IRD Schema and the IRD. The Minimal IRD Schema is described in detail in Section 10 of Chapter 1. Other Modules of the IRDS standard may prescribe additions to the Minimal IRD Schema. For example, the Basic Functional IRD Schema, provided as Module 2 of this standard, contains a sufficient number of entity-types, relationship-types, and attribute-types to allow an installation to use the IRDS immediately in a meaningful manner.

Another aspect of the IRD Schema is to be noted in reviewing Section 9 of Chapter 1. This section illustrates that the E-R model being used for these specifications is strongly typed. In particular, any object in the IRD or in the IRD Schema is always of a single type.

In Section 9 of Chapter 1 a detailed description of each of the types of IRD Schema entities, relationships, and attributes is specified. For the purposes of the present discussion, the diagram given in Figure 2 illustrates the association between the types of meta-entities (IRD Schema entities) via the types of meta-relationships (IRD Schema relationships). (The types in the rectangle defined by broken lines represent the extensions defined in Chapter 4.) In this figure, meta-entity-types are represented by squares, and meta-relationship-types by hexagons. By convention, each hexagon representing a meta-relationship-type is divided horizontally, with the top half containing an indicator of the name of the forward meta-relationship-type, and the bottom half containing an indicator of the name of the inverse meta-relationship-type. The cardinality (i.e., the number of allowable occurrences) of the meta-relationships of each type is also indicated. Note, specifically, that several of these meta-entity-types do not participate in any meta-relationships. These are IRD-PARTITION, QUALITY-INDICATOR, IRDS-LIMITS, IRDS-DEFAULTS, IRDS-RESERVED-NAMES, and NAMES. In this specification, the meta-entities associated with these particular meta-entity-types represent either conditions concerning entities, as is the case of IRD-PARTITION and QUALITY-INDICATOR, or they represent processing constraints associated with the IRDS and its interpretation of the IRD Schema content.

The concept of an IRD Schema is important for a number of reasons. Most important of these is the fact that this standard defines facilities which allow an installation to customize the IRD Schema to be responsive to its own requirements. This means that the installation can add those entity-types, relationship-types, attribute-types and attribute-group-types that will allow it to use the IRD as a repository for data that is of interest to that installation. A general description of the functionality necessary to extend the IRD Schema is provided below. Note that an installation is not allowed to modify the Minimal IRD Schema.



**Figure 2**  
**Associations of IRD Schema Entities**

### 3.4 Extending the IRD Schema

Assume that one wished to support installation requirements not supported by either the standard IRDS facilities or the installation's current IRD Schema. The installation would need to be able to modify the IRD Schema and would have to add new software facilities designed to operate against the modified IRD Schema. This is quite similar to adding a new application into a database environment. The principal difference is that, unlike a general database environment, the IRDS software assures the integrity of the existing and new IRD Schema and the IRD, based on "rules" which exist in the Minimal IRD Schema.

To illustrate how one would "extend" the IRD Schema to support additional requirements, it is necessary to understand the character of the IRD Schema. Basically, the IRD Schema can be defined in terms of entities, relationships and attributes. However, because of the potential for misunderstanding which may occur when discussing the contents of the IRD Schema and the IRD, similar, yet distinct, terminology is used to describe the IRD and the IRD Schema. In particular,

- (1) The IRD is assumed to contain:
  - (a) Entities.
  - (b) Relationships between entities.
  - (c) Attributes and attribute-groups which document characteristics of the entities and the relationships.
- (2) Similarly, the IRD Schema is assumed to contain:
  - (a) Entities, which are referred to as meta-entities.
  - (b) Relationships between meta-entities, which are referred to as meta-relationships.
  - (c) Attributes and attribute-groups, which are referred to as meta-attributes and meta-attribute-groups, which document characteristics of the meta-entities and the meta-relationships.

The commands that operate against the IRD Schema are specified in Subsection 5.1 of Chapter 1 of this standard, as well as in subsequent chapters. These IRD Schema maintenance and reporting commands operate only on meta-entities of the types represented by boxes in Figure 2. Thus, for example:

- (1) A new entity-type or a new relationship-type can be added to the IRD Schema using the "add meta-entity" command.
- (2) Existing attribute-type-validation-data or variation-names-data can be modified using the "modify meta-entity" command.
- (3) Attribute-types can be deleted from the IRD Schema using the "delete meta-entity" command.

After the new entity-type or relationship-type is added to the IRD Schema, it is normally desirable to associate attribute-types with that new IRD Schema element. This is accomplished using the "add meta-relationship" command; i.e., the fact that an attribute-type is associated with an entity-type or relationship-type is represented by a meta-relationship in the IRD Schema. Of course, these commands cannot violate any integrity rules associated with existing IRD Schema elements, or with the metadata in the IRD which is defined by those IRD Schema elements.

### 3.5 The Information Resource Dictionary

As noted above, the E-R data model is being used to describe the contents of the Information Resource Dictionary (IRD), the third level of the IRDS architecture. The result is that the IRD consists of a set of entities, relationships, and attributes where:

- (1) An entity represents a set of objects which are of interest to the user of the IRDS and which are being described in the IRD.
- (2) A relationship is a directed association between two IRD entities, and is used, in general, to describe an association between the sets of real-world objects which the entities represent.
- (3) The attributes of an entity or relationship represent properties of that entity or relationship.

An important aspect of how the E-R model is being used concerns the concept of "type". Each entity in the IRD has a unique type, which we refer to as the entity-type of the entity. Similarly, each relationship in the IRD has a unique type called the relationship-type of the relationship. For a given entity-type or relationship-type, there are a defined set of attribute-types and attribute-group-types (ordered sets of attribute-types) associated with the entity-type or relationship-type, and every attribute and attribute-group of an entity or relationship of the particular type must correspond to one of these attribute-types or attribute-group-types.

Commands are specified in Subsection 5.2 of Chapter 1 that allow a user of the IRDS to perform maintenance of the IRD and to report on its contents. Subsequent Modules may provide new commands or extensions to these. The structure of these commands is parallel to the ones operating on the IRD Schema.

### 3.6 The Information Resource Dictionary System

This standard specifies the syntax and semantics of a command language that operates against the IRD and the IRD Schema. The command language syntax is specified in a BNF-like form; the semantics are presented as a set of actions and rules. The semantics of a Panel (interactive) Interface are also specified. The Panel Interface specification identifies the content of an interactive interface, not the form of that interface. In both cases, the functionality is specified in terms of an E-R data model to facilitate understanding.

Module 1 presents the specification of basic functionality for operating on the IRD and the IRD Schema. Subsequent Modules specify extensions to this functionality.

## 4 Definition of a Conformant Implementation of the IRDS

This document provides a definition of the features that comprise the IRDS, as described in this American National Standard, hereafter referred to as the standard IRDS. The standard IRDS consists of seven Modules. Each Module of this standard is specified in a subsequent chapter of this document.

All conformant IRDSs shall supply Module 1, the Core standard. The remaining six Modules are optional, subject to the interrelationships and dependencies described below.

In developing the specifications for a standard IRDS, it was recognized that IRDS technology is still evolving and that uses for IRDS technology are continually expanding. Thus, it was necessary to provide a mechanism that allowed the establishment of a standard that would evolve as IRDS technology evolves. In addition, it was recognized that any one installation wanting to purchase an IRDS might not need all of the functionality that could be provided by an extended IRDS; this is particularly apparent when one examines the potential implementation environments for an IRDS, including the microcomputer environment. This approach allows the optional Modules to be obtained separately as users encounter the need for additional functionality. The following paragraphs identify the criteria that shall be met for a valid claim to be made that an implementation conforms to a standard IRDS.

### 4.1 Definition of the Standard IRDS

The standard IRDS consists of:

#### Module 1 -- The Core

This standard IRDS Module consists of the IRD Schema, the Minimal IRD Schema, and either the Command Language Interface or the Panel Interface, or both. The IRDS commands supported by the standard IRDS are:

- (1) The IRD Schema maintenance commands (Chapter 1, Subsection 5.1.1).
- (2) The IRD Schema output command (Chapter 1, Subsection 5.1.2).
- (3) The IRD maintenance commands (Chapter 1, Subsection 5.2.1).
- (4) The IRD output commands (Chapter 1, Subsection 5.2.2).
- (5) The IRD-IRD Interface commands (Chapter 1, Subsection 5.3.1).

The IRD-IRD Interface specification within this standard includes only the commands and clauses required for exporting and importing IRDS database content (these are identified by **\*\*EX/IM\*\***). The specification does not provide the standard notation and encoding rules for the data interchange. The notation and encoding rules, which will be based on Abstract Syntax Notation One (ASN.1) standards (ISO 8824 and 8825), are being developed, and are referred to as the IRDS Export/Import File Format standard.

A conformant implementation of this standard need not provide the IRD-IRD Interface specific commands and clauses until the IRDS standard for notation and encoding rules is available.



- (6) The utility commands:
- (a) The `set-session-defaults-command` (Chapter 1, Subsection 5.3.2.1).
  - (b) The `session-status-command` (Chapter 1, Subsection 5.3.2.2).
  - (c) The `help-command` (Chapter 1, Subsection 5.3.2.3).
  - (d) The `exit-IRD-system-command` (Chapter 1, Subsection 5.3.2.4).

With respect to the Panel Interface, the functionality for all of the above shall be required except for the following, in the case where the Command Language Interface is not also provided:

- (1) For the IRD output commands (Chapter 1, Subsection 5.2.2), in the event that the Command Language Interface is not provided in conjunction with the Panel Interface, implementation of the `output-syntax-panel-tree` is not required on the part of the implementor.
- (2) The functionality of the utility commands, with the exception of the `enter-panel-dialogue-command`. The functionality of the utility commands, however, are not made available via panel trees, as is the case with the other IRD commands. Rather, this functionality shall be provided through certain special features of the panel areas within the Panel Interface. The special features used in implementing the utility commands are:
  - (a) The Defaults option in the Action Area, which provides the functionality of the `set-session-defaults-command` (Chapter 1, Subsection 5.3.2.1).
  - (b) The State Area, which provides the functionality of the `session-status-command` (Chapter 1, Subsection 5.3.2.2).
  - (c) The Help Area, which provides the functionality of the `help-command` (Chapter 1, Subsection 5.3.2.3).
  - (d) The Exit option in the Action Area, which provides the functionality of the `exit-IRD-system-command` (Chapter 1, Subsection 5.3.2.4).

The panel areas mentioned above are identified in Section 8 of Chapter 1.

#### Module 2 -- The Basic Functional IRD Schema

This standard IRDS Module consists of the Basic Functional IRD Schema, an extension to the Minimal IRD Schema provided in Module 1 of the standard IRDS. Module 2 is dependent on the availability of Module 1.

#### Module 3 -- IRDS Security

This standard IRDS Module consists of the model and functionality of an access control facility that allows organizations to restrict access to IRD and IRD Schema functionality and content. This Module modifies and extends the IRD Schema, Command Language, and Panel Interface of Module 1 to support control of access to the IRD and its IRD Schema. Module 3 is dependent on the availability of Module 1.

#### Module 4 -- Extensible Life Cycle Phase Facility

This standard IRDS Module provides the basis for life cycle management of the contents of the IRD. The Module extends both the IRD Schema and the IRDS functionality of Module 1 to effect life cycle management. Module 4 is dependent on the availability of Module 1.

### Module 5 -- Procedure Facility

This standard IRDS Module provides a mechanism for defining and executing procedures composed of IRDS commands. The Module modifies and extends the IRD Schema of Module 1. Module 5 is dependent on the availability of Module 1, and requires that the Command Language syntax of Module 1 shall be available to the user of the procedure facility.

### Module 6 -- Application Program Interface

This standard IRDS Module consists of an interface to an implementation of the standard IRDS. This interface is invoked by providing Command Language syntax through the "Call" feature of any standard Language. The Module modifies the IRD Schema of Module 1. Module 6 is dependent on the availability of Module 1, and requires that the Command Language syntax of Module 1 shall be available to the user of this interface.

### Module 7 -- Entity Lists

This standard IRDS Module provides the capability for a name to be assigned to a list of entities, and specifies commands and panels which can be used to manipulate these lists of entities.

For each Module:

- (1) There shall be no requirement to implement the IRD Schema in an entity-relationship model, but the implementation shall be able to represent the IRD Schema as defined in Section 9 of Module 1.
- (2) Except as noted, i.e., Modules 4 and 6, an implementation shall be considered to be conformant with the standard IRDS if it uses EITHER the Command Language Interface or the Panel (screen-oriented) Interface, or BOTH.
- (3) The Command Language Interface assumes the specified syntax and semantics are implemented; i.e., the Command Language is a "concrete" syntax. The Panel Interface assumes the specific semantics of commands, but not the syntax; i.e., the Panel Interface treats the Command Language as an "abstract" syntax.

## **4.2 Definition of a Conformant Implementation**

A conformant implementation of a Module of a standard IRDS shall fully support the items identified in Subsection 4.1. A conformant implementation with the Command Language Interface shall accept the syntax and provide the specified semantics for each standard IRDS language element specified in the Module. A conformant implementation with the Panel Interface shall support the semantics of the commands specified in the Module.

### **4.2.1 Standard/Extended Mode**

There are two modes of operation for the standard IRDS:

- |                |   |
|----------------|---|
| Standard mode: | The IRDS shall perform as specified by the standard.  |
| Extended mode: | The IRDS shall perform all functionality defined by correct syntax and semantics, as specified in the standard. This means that functionality which is correct in Standard mode produces the same |

result in Extended mode. The implementor may provide extensions to the standard.

An implementation of the IRDS shall be conformant if either:

- (1) It operates in Standard mode, or
- (2) It operates in Extended mode, but can, at the option of the installation, be restricted to operate in Standard mode.

#### **4.2.2 Substitute or Additional Language Elements in the Command Language Interface**

An implementation shall not require the inclusion of substitute or additional language elements to accomplish a function identical to that of the standard IRDS language element. Additionally, throughout the standard IRDS, there are additional language elements whose syntax or function is specified to be, in part, implementor-defined. While the implementor specifies the constraints on that portion of each language element's syntax or rules that is indicated to be implementor-defined, such constraints shall not include any requirements for the inclusion of substitute or additional language elements. An implementation that includes language elements in addition to those in the standard IRDS meets the requirements of the IRDS standard. However, use of such extensions shall be optional. Documentation associated with an implementation shall identify any extensions that are included in the implementation.

#### **4.2.3 Character Substitution in the Command Language**

The definition of the IRDS character set (Chapter 1, Subsection 4.1) presents the complete character set for the standard IRDS. When an implementation does not provide for a graphic representation for all members of the IRDS character set, substitute graphic representations may be specified by the implementor to replace the characters not provided.

### **4.3 Relationship of a Conformant Program to a Conformant Implementation**

When the standard IRDS does not specify limits on parameters, such as the number of operands permitted in certain statements, the implementor determines the limits and the implementation shall be conformant, regardless of the limits selected. It is recognized that these limits will vary from one implementation to another, and may adversely affect transportability of software across conformant implementations.

### **4.4 Relationship of a Conformant IRDS Implementation to a System Environment**

This standard does not endorse any specific system environment. An implementation may be conformant to this standard alone, or it may be conformant to this standard in conjunction with a specification for another component of this environment. In the latter case, the implementation shall process statements according to the other component's specification, and if those statements access the IRD, they shall be defined in terms of the interfaces specified by this standard. In all cases, the result of any IRD action shall be as if the IRDS interfaces were separated and processed individually.

A binding may occur between components on either side of an interface defined by this standard. The method and timing of such binding is implementor defined.

If any function provided outside the IRDS implementation accomplishes a function specified by a standard IRDS, then the implementation shall not require the specification of the external function in place of, or in addition to, that standard IRDS language element's syntax or semantics.

#### **4.5 Scope of the Panel Interface**

For each Module, the standard IRDS Panel Interface shall provide the same functionality as that found in the Command Language Interface. In the case where a particular installation has implemented both a Panel Interface and a Command Language Interface, the Panel Interface shall also provide a "command option" which can be used to gain access to the Command Language Interface. Thus, selection of the command option allows access to the command language syntax for those users with appropriate permissions.

**Chapter 1****Module 1 - Core Standard****CONTENTS**

<b>SECTION</b>	<b>PAGE</b>
1 Scope, Purpose, and Application . . . . .	1-1
1.1 Scope . . . . .	1-1
1.2 Conformance . . . . .	1-1
1.3 Organization . . . . .	1-1
1.4 Notation . . . . .	1-2
1.5 Implementation Options . . . . .	1-4
1.6 Special Purpose Characters . . . . .	1-5
1.7 Messages . . . . .	1-6
2 References . . . . .	1-6
3 Definitions . . . . .	1-7
4 Basic Elements . . . . .	1-16
4.1 Characters . . . . .	1-16
4.2 Literals . . . . .	1-17
4.3 Tokens . . . . .	1-22
4.4 Entity and Meta-Entity Names . . . . .	1-26
4.5 Attributes . . . . .	1-31
4.6 Type Designators . . . . .	1-33
4.7 IRD Schema Language Tokens . . . . .	1-35
4.8 Security Tokens . . . . .	1-39
4.9 IRDS Functions . . . . .	1-40
4.10 The IRDS Data Model . . . . .	1-42
4.11 The IRD Schema Definition . . . . .	1-44
4.12 Reserved Words . . . . .	1-44
5 Command Specifications . . . . .	1-46
5.1 IRD Schema Commands . . . . .	1-48
5.1.1 IRD Schema Maintenance Commands . . . . .	1-49
5.1.1.1 Add Meta-Entity Command . . . . .	1-50
5.1.1.2 Modify Meta-Entity Command . . . . .	1-52
5.1.1.3 Delete Meta-Entity Command . . . . .	1-56
5.1.1.4 Add Meta-Relationship Command . . . . .	1-57
5.1.1.5 Modify Meta-Relationship Command . . . . .	1-59
5.1.1.6 Delete Meta-Relationship Command . . . . .	1-61
5.1.1.7 Modify Meta-Entity Access-Name Command . . . . .	1-62
5.1.1.8 Modify Meta-Entity Descriptive-Name Command . . . . .	1-64
5.1.1.9 Modify Meta-Entity Life-Cycle-Phase Command . . . . .	1-66
5.1.1.10 Copy Meta-Entity Command . . . . .	1-70
5.1.1.11 Deactivate IRD Command . . . . .	1-73
5.1.1.12 Restore IRD Schema Command . . . . .	1-74
5.1.1.13 Activate IRD Command . . . . .	1-75

SECTION	PAGE
5.1.2	IRD Schema Output Command . . . . . 1-81
5.2	IRD Commands . . . . . 1-82
5.2.1	IRD Maintenance Commands . . . . . 1-83
5.2.1.1	Add Entity Command . . . . . 1-85
5.2.1.2	Modify Entity Command . . . . . 1-87
5.2.1.3	Delete Entity Command . . . . . 1-89
5.2.1.4	Add Relationship Command . . . . . 1-90
5.2.1.5	Modify Relationship Command . . . . . 1-91
5.2.1.6	Delete Relationship Command . . . . . 1-92
5.2.1.7	Modify Entity-Access-Name Command . . . . . 1-93
5.2.1.8	Modify Entity-Descriptive-Name Command . . . . . 1-94
5.2.1.9	Modify Entity Life-Cycle-Phase Command . . . . . 1-96
5.2.1.10	Copy Entity Command . . . . . 1-97
5.2.2	IRD Output Commands . . . . . 1-99
5.2.2.1	General Output Command . . . . . 1-102
5.2.2.2	Output Impact-of-Change Command . . . . . 1-103
5.2.2.3	Output Syntax Command . . . . . 1-105
5.3	General Commands . . . . . 1-108
5.3.1	IRD-IRD Interface Commands . . . . . 1-109
5.3.1.1	Create IRD Command . . . . . 1-109
5.3.1.2	Export IRD Command . . . . . 1-111
5.3.1.3	Check IRD Schema Compatibility Command . . . . . 1-114
5.3.1.4	Import IRD Command . . . . . 1-119
5.3.2	Utility Commands . . . . . 1-120
5.3.2.1	Set Session Defaults Command . . . . . 1-121
5.3.2.2	Session Status Command . . . . . 1-123
5.3.2.3	Help Command . . . . . 1-124
5.3.2.4	Exit IRDS Command . . . . . 1-126
5.3.2.5	Enter Panel Dialogue Command . . . . . 1-126
6	Command-Clause Specifications . . . . . 1-127
6.1	Entity-Type Qualification Command-Clause . . . . . 1-127
6.2	Entity-Type Command-Clause . . . . . 1-128
6.3	Simple Attribute Command-Clause . . . . . 1-128
6.4	New Repeating Attribute Command-Clause . . . . . 1-130
6.5	New Simple Attribute-Group Command-Clause . . . . . 1-132
6.6	New Repeating Attribute-Group Command-Clause . . . . . 1-134
6.7	New Text Attribute Command-Clause . . . . . 1-136
6.8	Modified Repeating Attribute Command-Clause . . . . . 1-138
6.9	Modified Simple Attribute-Group Command-Clause . . . . . 1-141
6.10	Modified Repeating Attribute-Group Command-Clause . . . . . 1-144
6.11	Modified Text Attribute Command-Clause . . . . . 1-148
6.12	Entity-Access-Name Selection Command-Clause . . . . . 1-153
6.17	Order Command-Clause . . . . . 1-154
6.18	Relationship-Type Qualification Command-Clause . . . . . 1-154
6.19	Start Line Number Command-Clause . . . . . 1-155
6.20	Line Number Increment Command-Clause . . . . . 1-156
6.21	Line Range Command-Clause . . . . . 1-156
6.22	New Entity Attributes Command-Clause . . . . . 1-158
6.23	Modified Entity Attributes Command-Clause . . . . . 1-158

SECTION	PAGE	
6.24	Relationship Identification Command-Clause . . . . .	1-159
6.25	New Relationship Identification Command-Clause . . . . .	1-160
6.26	Order Restriction Command-Clause . . . . .	1-163
6.27	Entity-Descriptive-Name Declaration Command-Clause . . . . .	1-164
6.28	New Version Command-Clause . . . . .	1-164
6.29	Text Attribute Resequenece Command-Clause . . . . .	1-165
6.30	Quality Indicator Designation Command-Clause . . . . .	1-167
6.31	Quality Indicator Restriction Command-Clause . . . . .	1-167
6.32	With Relationships Command-Clause . . . . .	1-168
6.33	Entity Selection Criteria Command-Clause . . . . .	1-168
6.34	Entity-Type Restriction Command-Clause . . . . .	1-171
6.35	Relationship Existence Restriction Command-Clause . . . . .	1-172
6.36	Attribute Restriction Command-Clause . . . . .	1-173
6.37	Attribute-Group Restriction Command-Clause . . . . .	1-175
6.38	Using IRD-Views Command-Clause . . . . .	1-177
6.40	Text Attribute Substring Restriction Command-Clause . . . . .	1-178
6.41	Entity-Descriptive-Name Selection Command-Clause . . . . .	1-179
6.42	All Entities Command-Clause . . . . .	1-180
6.43	New Relationship Attributes Command-Clause . . . . .	1-180
6.44	Modified Relationship Attributes Command-Clause . . . . .	1-181
6.45	Relationship Selection Command-Clause . . . . .	1-182
6.46	IRD Life-Cycle-Phase Restriction Command-Clause . . . . .	1-184
6.47	Audit Attribute Restriction Command-Clause . . . . .	1-184
6.48	Related Entities Command-Clause . . . . .	1-186
6.49	Alternate-Name Restriction Command-Clause . . . . .	1-188
6.50	Relationship-Type Restriction Command-Clause . . . . .	1-189
6.51	In IRD Command-Clause . . . . .	1-191
6.52	In File Command-Clause . . . . .	1-192
6.53	IRD Schema Source Command-Clause . . . . .	1-193
6.54	Load IRD Command-Clause . . . . .	1-193
6.55	IRD Schema Export File Command-Clause . . . . .	1-194
6.56	IRD Export File Command-Clause . . . . .	1-195
6.57	Location Command-Clause . . . . .	1-195
6.58	File Title Suffix Command-Clause . . . . .	1-196
6.59	Other IRD Schema Command-Clause . . . . .	1-196
6.60	Exclude Relationships of Type Command-Clause . . . . .	1-197
6.61	IRD Life-Cycle-Phase Designation Command-Clause . . . . .	1-197
6.62	Sort Command-Clause . . . . .	1-198
6.63	Show Predefined Display Command-Clause . . . . .	1-199
6.64	Route Command-Clause . . . . .	1-200
6.65	Show Title Command-Clause . . . . .	1-201
6.66	Entity-Type Show Restriction Command-Clause . . . . .	1-202
6.67	Qualified Show Command-Clause . . . . .	1-203
6.68	Show IRD Life-Cycle-Phase Command-Clause . . . . .	1-204
6.69	Show Entity-Access-Name Command-Clause . . . . .	1-205
6.70	Show Entity-Descriptive-Name Command-Clause . . . . .	1-206
6.71	Show Entity-Type Command-Clause . . . . .	1-206
6.72	Show Relationships Command-Clause . . . . .	1-207
6.73	Show Attributes Command-Clause . . . . .	1-209
6.74	Show All Command-Clause . . . . .	1-212
6.75	Show Relationship Syntax Command-Clause . . . . .	1-213

SECTION	PAGE
6.77	IRDS Function Restriction Command-Clause . . . . . 1-214
6.78	Entity Assigned-Access-Name Restriction Command-Clause . . . . . 1-215
6.79	Entity Assigned-Descriptive-Name Restriction Command-Clause . . . . . 1-215
6.80	Variation-Name Restriction Command-Clause . . . . . 1-216
6.81	Revision-Number Restriction Command-Clause . . . . . 1-216
6.82	Show Counts Command-Clause . . . . . 1-217
6.83	Show Quality-Indicator Command-Clause . . . . . 1-218
6.101	Meta-Entity-Type Command-Clause . . . . . 1-219
6.103	New Meta-Attributes Command-Clause . . . . . 1-219
6.104	Modified Meta-Attributes Command-Clause . . . . . 1-220
6.106	Simple Meta-Attribute Command-Clause . . . . . 1-221
6.107	New Repeating Meta-Attribute Command-Clause . . . . . 1-223
6.108	New Meta-Attribute-Group Command-Clause . . . . . 1-224
6.109	New Text Meta-Attribute Command-Clause . . . . . 1-225
6.110	Modified Repeating Meta-Attribute Command-Clause . . . . . 1-227
6.111	Modified Meta-Attribute-Group Command-Clause . . . . . 1-230
6.112	Modified Text Meta-Attribute Command-Clause . . . . . 1-234
6.113	Sort Meta-Entities Command-Clause . . . . . 1-237
6.114	IRD Schema Show All Command-Clause . . . . . 1-238
6.115	Show Meta-Attributes Command-Clause . . . . . 1-239
6.116	Show Meta-Relationships Command-Clause . . . . . 1-241
6.117	Show Related Meta-Entities Command-Clause . . . . . 1-241
6.118	Meta-Entity Selection Command-Clause . . . . . 1-242
6.119	Meta-Entity-Type Restriction Command-Clause . . . . . 1-244
6.120	Meta-Entity-Type Qualification Command-Clause . . . . . 1-245
6.121	Meta-Relationship-Type Qualification Command-Clause . . . . . 1-246
6.122	Meta-Relationship Existence Restriction Command-Clause . . . . . 1-246
6.123	Meta-Attribute-Group Restriction Command-Clause . . . . . 1-247
6.124	Meta-Attribute Restriction Command-Clause . . . . . 1-248
6.125	Text Meta-Attribute Resequence Command-Clause . . . . . 1-249
6.126	Meta-Entity Descriptive-Name Command-Clause . . . . . 1-251
6.127	New Meta-Entity Version Command-Clause . . . . . 1-251
6.128	With Meta-Relationships Command-Clause . . . . . 1-252
6.129	IRD Schema Life-Cycle-Phase Restriction Command-Clause . . . . . 1-252
6.130	Show IRD Schema Life-Cycle-Phase Command-Clause . . . . . 1-252
6.132	Using IRD-Schema-Views Command-Clause . . . . . 1-253
7	Cross Reference Indices . . . . . 1-254
7.1	Alphabetic Command Index . . . . . 1-254
7.2	Alphabetic Command-Clause Index . . . . . 1-255
7.3	Error and Warning Condition Index . . . . . 1-257
8	Panel Interface . . . . . 1-272
8.1	IRD Schema Panel Areas . . . . . 1-275
8.1.1	State Area . . . . . 1-276
8.1.2	Data Area . . . . . 1-276
8.1.3	Action Area . . . . . 1-277
8.1.4	Message Area . . . . . 1-278
8.1.5	Help Area . . . . . 1-278
8.2	IRD Schema Panel Trees . . . . . 1-279
8.2.1	IRD Schema Maintenance Panel Trees . . . . . 1-280



SECTION		PAGE
	8.2.1.1 Add Meta-Entity Panel Tree . . . . .	1-281
	8.2.1.2 Modify Meta-Entity Panel Tree . . . . .	1-282
	8.2.1.3 Delete Meta-Entity Panel Tree . . . . .	1-283
	8.2.1.4 Add Meta-Relationship Panel Tree . . . . .	1-283
	8.2.1.5 Modify Meta-Relationship Panel Tree . . . . .	1-285
	8.2.1.6 Delete Meta-Relationship Panel Tree . . . . .	1-286
	8.2.1.7 Modify Meta-Entity-Access-Name Panel Tree . . . . .	1-287
	8.2.1.8 Modify Meta-Entity-Descriptive-Name Panel Tree . . . . .	1-288
	8.2.1.9 Modify Meta-Entity Life-Cycle-Phase Panel Tree . . . . .	1-289
	8.2.1.10 Copy Meta-Entity Panel Tree . . . . .	1-289
	8.2.1.11 Deactivate IRD Panel Tree . . . . .	1-290
	8.2.1.12 Restore IRD Schema Panel Tree . . . . .	1-291
	8.2.1.13 Activate IRD Panel Tree . . . . .	1-291
	8.2.2 IRD Schema Output Panel Trees . . . . .	1-292
	8.2.2.1 IRD Schema Output Panel Tree . . . . .	1-293
	8.2.2.2 Inquire Meta-Entity Panel Tree . . . . .	1-294
	8.2.2.3 Inquire Meta-Relationship Panel Tree . . . . .	1-294
8.3	IRD Panel Areas . . . . .	1-295
	8.3.1 State Area . . . . .	1-296
	8.3.2 Data Area . . . . .	1-296
	8.3.3 IRD Schema Area . . . . .	1-297
	8.3.4 Action Area . . . . .	1-297
	8.3.5 Message Area . . . . .	1-298
	8.3.6 Help Area . . . . .	1-299
8.4	IRD Panel Trees . . . . .	1-299
	8.4.1 IRD Maintenance Panel Trees . . . . .	1-300
	8.4.1.1 Add Entity Panel Tree . . . . .	1-301
	8.4.1.2 Modify Entity Panel Tree . . . . .	1-302
	8.4.1.3 Delete Entity Panel Tree . . . . .	1-303
	8.4.1.4 Add Relationship Panel Tree . . . . .	1-304
	8.4.1.5 Modify Relationship Panel Tree . . . . .	1-305
	8.4.1.6 Delete Relationship Panel Tree . . . . .	1-306
	8.4.1.7 Modify Entity-Access-Name Panel Tree . . . . .	1-307
	8.4.1.8 Modify Entity-Descriptive-Name Panel Tree . . . . .	1-308
	8.4.1.9 Modify Entity Life-Cycle-Phase Panel Tree . . . . .	1-309
	8.4.1.10 Copy Entity Panel Tree . . . . .	1-310
	8.4.2 IRD Output Panel Trees . . . . .	1-311
	8.4.2.1 General Output Panel Tree . . . . .	1-312
	8.4.2.2 Output Impact-of-Change Panel Tree . . . . .	1-312
	8.4.2.3 Output Syntax Panel Tree . . . . .	1-313
	8.4.2.4 Inquire Entity Panel Tree . . . . .	1-314
	8.4.2.5 Inquire Relationship Panel Tree . . . . .	1-314
8.5	IRD-IRD Interface Panel Trees . . . . .	1-315
	8.5.1 Create IRD Panel Tree . . . . .	1-316
	8.5.2 Export IRD Panel Tree . . . . .	1-317
	8.5.3 Check IRD Schema Compatibility Panel Tree . . . . .	1-318
	8.5.4 Import IRD Panel Tree . . . . .	1-318
9	The IRD Schema . . . . .	1-319
	9.1 Meta-Entity Types . . . . .	1-319
	9.2 Meta-Relationship Types . . . . .	1-321

SECTION	PAGE	
9.2.1	Unique and Nonunique Meta-Relationship Types . . . . .	1-321
9.2.2	Sequenced and Nonsequenced Meta-Relationship Types . . . . .	1-322
9.2.3	Meta-Relationship Type Ratios . . . . .	1-322
9.2.4	The IRD Schema Life-Cycle-Phase Integrity Rule . . . . .	1-323
9.2.5	Meta-Relationship Type Definitions . . . . .	1-323
9.3	Meta-Attribute Types . . . . .	1-326
9.4	Meta-Attribute-Types for Meta-Entity Types . . . . .	1-347
9.5	Meta-Attribute Types for Meta-Relationship Types . . . . .	1-347
9.6	Meta-Attribute Types for Meta-Attribute Group Types . . . . .	1-347
9.7	Meta-Attribute Group Types for Meta-Entity Types . . . . .	1-352
9.8	The IRD Schema Change Control Facility . . . . .	1-352
9.9	The Definition of IRD Partitions in the IRD Schema . . . . .	1-355
10	The Minimal IRD Schema . . . . .	1-355
10.1	Purpose . . . . .	1-356
10.2	Minimal IRD Schema Descriptors . . . . .	1-356
10.2.1	Entity Types in the Minimal IRD Schema . . . . .	1-357
10.2.1.1	IRDS User . . . . .	1-357
10.2.1.2	IRD View . . . . .	1-358
10.2.1.3	IRD Schema View . . . . .	1-358
10.2.2	Relationship Types in the Minimal IRD Schema . . . . .	1-358
10.2.2.1	IRDS User Has IRD View . . . . .	1-359
10.2.2.2	IRDS User Has IRD Schema View . . . . .	1-359
10.2.3	Relationship Class Types in the Minimal IRD Schema . . . . .	1-360
10.2.3.1	Has . . . . .	1-360
10.2.4	Attribute Types in the Minimal IRD Schema . . . . .	1-361
10.2.4.1	Added By . . . . .	1-362
10.2.4.2	Default View . . . . .	1-362
10.2.4.3	IRD Partition Name . . . . .	1-362
10.2.4.4	Last Modified By . . . . .	1-363
10.2.4.5	Number of Times Modified . . . . .	1-363
10.2.4.6	IRD Schema Phase Name . . . . .	1-363
10.2.4.7	System Date . . . . .	1-363
10.2.4.8	System Generated Name . . . . .	1-364
10.2.4.9	System Time . . . . .	1-364
10.2.5	Attribute Group Types in the Minimal IRD Schema . . . . .	1-365
10.2.5.1	Date Time Added . . . . .	1-366
10.2.5.2	Date Time Last Modified . . . . .	1-366
10.2.6	Attribute Type Validation Data in the Minimal IRD Schema . . . . .	1-366
10.2.6.1	Yes or No Value . . . . .	1-367
10.2.7	Attribute Type Validation Procedures in the Minimal IRD Schema . . . . .	1-367
10.2.7.1	Range Validation . . . . .	1-368
10.2.7.2	Value Validation . . . . .	1-368
10.2.8	IRD Partitions in the Minimal IRD Schema . . . . .	1-368
10.2.8.1	Archived Life-Cycle-Phase . . . . .	1-369
10.2.8.2	Controlled Life-Cycle-Phase . . . . .	1-369
10.2.8.3	Security . . . . .	1-370
10.2.8.4	Uncontrolled Life-Cycle-Phase . . . . .	1-370
10.2.9	IRDS Defaults in the Minimal IRD Schema . . . . .	1-370
10.2.9.1	Existing IRDS Defaults . . . . .	1-370
10.2.10	IRDS Limits in the Minimal IRD Schema . . . . .	1-371

SECTION	PAGE
10.2.10.1 Existing IRDS Limits . . . . .	1-371
10.2.11 IRDS Reserved Names in the Minimal IRD Schema . . . . .	1-372
10.2.11.1 Standard Reserved Names . . . . .	1-372
10.2.12 Names in the Minimal IRD Schema . . . . .	1-372
10.2.12.1 Naming Rules . . . . .	1-372
10.3 Meta-Relationships in the Minimal IRD Schema . . . . .	1-373
10.3.1 Relationship Type Connects Entity Type . . . . .	1-373
10.3.2 Relationship Type Member of Relationship Class Type . . . . .	1-373
10.3.3 Entity Type Contains Attribute Type . . . . .	1-373
10.3.4 Entity Type Contains Attribute Group Type . . . . .	1-373
10.3.5 Relationship Type Contains Attribute Type . . . . .	1-373
10.3.6 Relationship Type Contains Attribute Group Type . . . . .	1-373
10.3.7 Attribute Group Type Contains Attribute Type . . . . .	1-374
10.3.8 Attribute Type Uses Attribute Type Validation Data . . . . .	1-375
10.3.9 Relationships and Their Attributes . . . . .	1-375
11 The Information Resource Dictionary (IRD) . . . . .	1-375
11.1 Entities and Their Attributes . . . . .	1-375
11.2 Relationships and Their Attributes . . . . .	1-376
Appendix - Collected Syntax . . . . .	1-377
Figures	
Figure 1 - Meta-entity-Types and Meta-Relationship-Types in the IRD Schema . . . . .	1-45
Tables	
Table 1 - Suggested Review Sequence . . . . .	1-3
Table 2 - Meta-Attribute-Type / Meta-Entity-Type Associations (Page 1) . . . . .	1-348
Table 2 - Meta-Attribute-Type / Meta-Entity-Type Associations (Page 2) . . . . .	1-349
Table 2 - Meta-Attribute-Type / Meta-Entity-Type Associations (Page 3) . . . . .	1-350
Table 3 - Meta-Attribute-Type / Meta-Relationship-Type Associations . . . . .	1-351
Table 4 - Meta-Attribute-Group-Type / Meta-Attribute-Type Associations . . . . .	1-351
Table 5 - Meta-Entity-Type / Meta-Attribute-Group-Type Associations . . . . .	1-353
Table 6 - Entity-Type / Attribute-Type Associations . . . . .	1-374
Table 7 - Entity-Type / Attribute-Group-Type Associations . . . . .	1-374



## **Chapter 1**

### **Module 1 - Core Standard**

#### **1 Scope, Purpose, and Application**

##### **1.1 Scope**

This Module of the standard IRDS consists of six major components:

- (1) Basic Definitions.
- (2) A Command Language.
- (3) A Panel Interface.
- (4) The IRD Schema.
- (5) The Minimal IRD Schema.
- (6) The Information Resource Dictionary.

##### **1.2 Conformance**

The Definition of a Conformant Implementation of Module 1 of a standard IRDS is given in the Definition of a Conformant Implementation of the IRDS (see Section 4 of Requirements for a Conformant Implementation).

##### **1.3 Organization**

Basic definitions are given in Section 3.

The formal specification of the command language syntax is organized as follows:

- (1) Basic elements are defined in Section 4.
- (2) Command specifications are given in Section 5.
- (3) Command-clause specifications are given in Section 6.

All maintenance and retrieval functions of the standard IRDS are supported through the command language. The command language shall be executable interactively and in batch mode, although some commands are restricted to one mode or the other.

In the command language specifications, three types of rules are given: syntax rules, general rules, and security rules. Syntax rules are those which determine whether the commands and command-clauses have correct format. General rules are those which enforce integrity constraints, determine consistency, and control the operation of commands. Security rules are those which determine whether specific actions are authorized for a given user.

Each basic element is described in the following fashion:

- (1) Its function within the language is defined.
- (2) Its format is defined.
- (3) Syntax rules are given.
- (4) General rules are given.
- (5) Error and warning conditions are defined.

Command specifications have the following format:

- (1) The function of the command is defined.
- (2) Its format is defined.
- (3) Syntax rules are given.
- (4) General rules are given.
- (5) Security rules are given.
- (6) Actions taken by the system are given.
- (7) Error and warning conditions are defined.

Note that the commands are grouped hierarchically by the general types of functions which they perform. Each such grouping also has the same format so that common rules can be defined for the group. Where there are several levels of grouping, rules for the higher levels also apply to lower levels.

Each command-clause is defined as follows:

- (1) Its function within the language is defined.
- (2) Its format is defined.
- (3) Syntax rules are given.
- (4) General rules are given.
- (5) Error and warning conditions are defined.

A command-clause can be composed of other command-clauses, and a given command-clause can be a component of several command-clauses. Consequently, each command-clause definition occurs at the same level in the numbering scheme of this document.

Section 7 provides alphabetic indices for the commands and command-clauses discussed in Sections 5 and 6.

The specifications for the Panel Interface are presented in Section 8.

Sections 9 and 10 provide the specification of the IRD Schema and the Minimal IRD Schema, respectively.

Section 11 specifies the Information Resource Dictionary (IRD) which shall be made available within an implementation of the standard IRDS.

The suggested order of review for this chapter is given in Table 1.

#### **1.4 Notation**

The syntax of this language is described using a simple variant of Backus-Naur-Form (BNF). In particular,

- (1) Lowercase words, some containing hyphens, shall be used to denote syntactic categories. For example:

**Table 1**  
**Suggested Review Sequence**

ORDER	SECTION	TITLE	ACTION
1	1	SCOPE, PURPOSE AND APPLICATION	READ
2	2 - 4	REFERENCES, DEFINITIONS, AND BASIC DEFINITIONS	SKIM
3	5	COMMAND SPECIFICATIONS	READ
4	10	THE MINIMAL IRD SCHEMA	SKIM
5	5.2	IRD COMMANDS	READ
6	9	THE IRD SCHEMA	SKIM
7	5.1	IRD SCHEMA COMMANDS	READ
8	5.3	GENERAL COMMANDS	READ
9	6	COMMAND-CLAUSES	AS REQUIRED
10	8	PANEL INTERFACE	READ
11	11	THE INFORMATION RESOURCE DICTIONARY	READ
12	7	CROSS REFERENCE INDICES	AS REQUIRED

attribute-clause  
spacing-character

- (2) Uppercase words shall be used to identify IRDS-defined words, for example:

ADD  
WITH ATTRIBUTES

IRDS-defined words shall not be reserved except as noted. They may be required at specific places in the language.

- (3) Square brackets, [ ], enclose optional items.

- (4) Braces, { }, enclose a repeating item. The item can repeat zero to  $n$  times, and repetitions move from left to right.
- (5) Comments and explanations of the syntax specification are delimited by a slash and asterisk at the beginning and an asterisk and slash at the end. For example:

xxxx-yyy ::= /\* explanation of xxxx-yyy \*/

defines xxxx-yyy within a comment.

- (6) When a term is expanded or defined in another section, a cross-reference to that Section is given on the far right-hand side. The lack of a cross-reference indicates that the term is expanded within the section.

### 1.5 Implementation Options

The following implementation options shall be available:

- (1) Hyphens are used as a spacing-character in this document. The implementor may substitute an underscore, "\_", or make this a session option. Thus, both of these representations of an IRDS-defined word are valid for different implementations:

ENTITY-TYPE, ENTITY\_TYPE

- (2) The use of lowercase letters for IRDS-defined words or IRDS-names may be restricted. IRDS-defined words are similar to keywords, but they are not reserved except as noted. They are referred to as "irds-word"s in the syntax specification. The term "IRDS-name", referred to as "irds-name" in the syntax specification, means the standard format of names in the IRD or IRD Schema.

Lowercase letters shall be equivalent with the corresponding uppercase letters for purposes of IRDS-defined words and irds-names. Thus, all of the following representations are equivalent representations of the same IRDS-defined word:

ENTITY; Entity; entity, enTity.

The implementation may specify maximum values for:

- (a) integers;
- (b) lengths of names;
- (c) lengths of lines of texts;
- (d) lengths of strings, and
- (e) number of lines of text.

The standard provides for minimal acceptable maxima for integers, length of names, and number of lines of text. It suggests criteria for selecting string lengths and provides for a minimum length for lines of text.

- (1) Certain command-clauses are implementation-dependent. For such command-clauses the syntax specification will define the command clause with the comment "/\* implementor-defined-format \*/". Here the implementor may define all rules pertaining to the command-clause.



- (2) In certain other cases, it is anticipated that the implementor may have need to define other command-clauses. The syntax specification notes these situations by using the symbol "implementor-defined-clauses". The anticipated usage of such command-clauses is discussed in the General Rules in each such case.

## 1.6 Special Purpose Characters

The following characters have special purposes within the grammar:

- (1) equals-sign
- (2) quotation marks
- (3) commas
- (4) parentheses
- (5) semicolon
- (6) colon
- (7) null-mark
- (8) asterisk
- (9) question mark
- (10) apostrophe

The equals-sign, =, shall be used to distinguish an IRD Schema descriptor from a user-supplied value. It is most commonly used for specifying attributes associated with either entities or relationships. In this case, the attribute-type's name is to the left of the equals-sign, and the value is to the right.

Quotation marks shall be used to delimit string-literals.

Commas shall be used to separate elements of a list. Their use shall be required when no other delimiter is able to indicate that another element follows within the list. Thus, they shall be required to separate values of a repeating attribute, but they are optional as separators of groups of attributes within a repeating attribute-group.

Parentheses shall be used for the following purposes:

- (1) To delimit an entity's version-identifier.
- (2) To delimit line number defaults for a new text attribute.
- (3) To delimit a range of line numbers within an existing text attribute.

The semicolon shall be used as a command terminator.

The colon shall be used to indicate concatenation. It has two uses:

- (1) In the version-identifier, it concatenates the variation-name with the revision-number.
- (2) It is used to qualify the names of shared attribute-types and attribute-group-types when used as input-parameters to IRDS Functions (see Subsection 4.9). Suppose *ET1* and *ET2* are entity-types sharing a common attribute-type *AT*. Then *ET1:AT* can be used to specify *AT* within *ET1*.

The null-mark shall be an implementation-selected special character. Its purpose is two-fold:

- (1) It is used to test for or identify a descriptor that does not exist.

- (2) It is used to indicate that the system shall assign a system-generated access-name to an entity in the add-entity-command.

The asterisk has two uses:

- (1) It is part of the comment delimiters /\* and \*/.
- (2) It can be used as a matching symbol in specifying search criteria. It designates any sequence of characters. Note that the asterisk does not match a null-mark, even though it matches a null sequence of characters.

The question mark can also be used in selection. It represents any single, non-null character. These characters shall not be used outside of a string-literal for other than their expressed purposes.

The apostrophe shall be used to denote an IRDS Function. IRDS Functions are syntactic devices which are used to return information about IRD and IRD Schema descriptors. Their usage is similar to "predefined attributes" found in several programming languages. They are used in IRD and IRD Schema output commands.

## 1.7 Messages

There shall be four levels of messages: information, warning, error, and severe error.

The wording of messages is not specified. This document only identifies warning and error conditions, which will correspond to warning and error messages. Each of these conditions is identified by a condition identifier. The condition identifier consists of a one character prefix followed by a five digit number. The prefix is either a "W", which identifies a warning condition, or an "E", which identifies an error condition. The first two digits identify the Module of the standard IRDS. The last three digits identify the condition number within the type of condition (i.e., conditions "E01001" and "W01001" are both allowed). Information messages and severe error conditions are regarded as implementation dependent.

The implementor shall provide a mechanism for the installation to customize the wording of messages.

## 2 References

This standard refers to the following International standards:

- (1) ISO 8824, Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)
- (2) ISO 8825, Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

### 3 Definitions

The following are definitions of terms used in this standard.

**access name.** In an IRDS, at IRD or IRD schema level, the primary identifier of each entity or meta-entity, i.e., the name by which the entity is known to the user. It consists of the assigned access name and the version identifier.

**alias.** An alternate name.

**alternate name.** An attribute-type of entity-types; instances are used to record in the IRD names by which entities are known outside the IRDS. The same alternate name may be associated with more than one entity.

**assigned access name.** (1) In an IRD, at data content level, a user- or system-assigned name which, in combination with variation name, provides unique access to an entity when first added to the IRD. (2) In an IRD, at schema level, a user-assigned name which provides unique access to a meta-entity when first added to the schema.

**assigned descriptive name.** In an IRDS, at IRD or IRD schema level, a name for an entity or meta-entity which is more descriptive than the assigned access name.

**attribute.** In an IRD, a property of an entity or relationship.

**attribute-group.** In an IRD, an ordered set of two or more attributes used together, e.g., date and time-of-day to form date-time.

**attribute-group-type.** In an IRDS, at IRD schema level, an ordered set of two or more attribute-types used together, e.g., LOW-OF-RANGE and HIGH-OF-RANGE.

**attribute-type.** In an IRDS, at IRD schema level, the label for a set of attributes which may be common to an entity-type or a relationship-type.

**audit.** In a database management system, the independent review and examination of system activities and records to test adequacy and effectiveness of data security and data integrity procedures, to ensure compliance with established policy and operational procedures, and to recommend any necessary changes.

**audit trail.** A chronological record of system activities which is sufficient to enable the reconstruction, review, and examination of the sequence of environments and activities surrounding or leading to each event in the path of a transaction from its inception to output of final results.

**Backus Naur form (BNF).** (1) A metalanguage used to specify or describe the syntax of a language, and in which each symbol, by itself, represents a set of strings of symbols. (2) Synonymous with Backus normal form.

**Backus normal form.** Synonym for Backus Naur form.

**Base Document.** (1) In TC X3H4, a functional specification of a base level (level 0) information resource dictionary system. This document may include a detailed description of functions to be supported, as well as the syntax and semantics of a language to be used in operation of the system. (2) See also Requirements Document.

**BNF.** (1) Backus Naur form. (2) Backus normal form.

**certification.** In computer systems, the technical evaluation, made as part of and in support of the accreditation process, that establishes the extent to which a particular computer system or network design and implementation meet a prespecified set of requirements.

**computer program.** A sequence of instructions suitable for processing by a computer. Processing may include the use of an assembler, a compiler, an interpreter, or a translator to prepare the program for execution, as well as to execute it.

**data.** Any representation of entities, relationships, or attributes such as characters or analog quantities to which meaning is or might be assigned.

**data administration.** (1) The responsibility for definition, organization, supervision, and protection of data. (2) See also data resource management.

**data base.** (1) A set of data, part or the whole of another set of data, consisting of at least one file, that is sufficient for a given purpose or for a given data processing system. (2) See also database.

**data definition language (DDL).** (1) In a database management system, the set of declarative statements that describe the contents of a database by defining and specifying all of the component data elements together with their relationships within a data structure. (2) A language for defining the logical and/or physical structure of a database.

**data dictionary.** (1) A subset of a data dictionary/directory that provides definitions for each data entity. (2) See also data dictionary/directory, information resource dictionary.

**data dictionary/directory (DD/D).** (1) An inventory of data resources that controls the totality of data entities within an application. It is the repository of all descriptive information about each data entity. (2) See also information resource dictionary.

**data dictionary/directory system (DD/DS).** (1) The computer software system that maintains and manages a data dictionary/directory. (2) See also information resource dictionary system.

**data dictionary system.** (1) A computer software system that maintains and manages a data dictionary. (2) See data dictionary/directory system, information resource dictionary system.

**data directory.** (1) A subset of a data dictionary/directory that identifies data location and ownership. (2) See also data dictionary/directory, information resource dictionary.

**data directory system.** (1) A computer software system that maintains and manages a data directory. (2) See data dictionary/directory system, information resource dictionary system.

**data element.** (1) A named unit of data. It can be used to describe the atomic level of data, whether computerized or manual, as viewed by a user. (2) In database usage, a named identifier of the entities and attributes that are represented in a database. (3) See element.

**data independence.** A characteristic of a database such that data structure or content can be changed without affecting computer application programs.

**data integrity.** (1) In information processing, the condition in which data is current, consistent, and accurate. (2) Contrast with system integrity.

**data management.** (1) The function of controlling the acquisition, analysis, storage, retrieval, and distribution of data. (2) See data resource management.

**data manipulation language (DML).** In a database management system, the set of statements that provide the capability for interaction with the database in order to perform certain basic functions necessary to maintain and use the database.

**data model.** A description of the organization of data in a manner that reflects the information structure of the enterprise. (2) See also data structure.

**data resource management.** (1) The responsibility for planning, organizing, and controlling data resources consistent with the overall goals and objectives of the enterprise. (2) See information resource management.

**data resources.** All manual and automated data used by an enterprise to represent its information.

**data security.** The protection of data from accidental or malicious modification or destruction and from accidental or intentional disclosure to unauthorized personnel.

**data structure.** (1) The logical relationships which exist among the units of data. (2) An instance or occurrence of a data model.

**database (DB).** (1) A large collection of interrelated data that is stored together to serve one or more applications. (2) In CODASYL, the data defined and described by one schema. (3) A repository for data sufficient to serve some purpose of an enterprise on a continuing basis. (4) See also data base.

**database administration.** The responsibility for definition, organization, supervision, and protection of a database.

**database management system (DBMS).** An integrated set of computer system programs that collectively provide the capabilities required for centralized management, that is, organization, access, and control of a database.

**database schema.** (1) The set of statements, expressed in a data definition language, which provides a complete description of a database, and which includes the names and definitions of all of the data elements and their associated attributes. (2) See also IRD schema.

**database subschema.** A subset of a schema which represents a view of a database from the perspective of a particular application or user.

**DB.** Database.

**DBMS.** Database management system.

**DD/D.** Data dictionary/directory.

**DDL.** Data description language.

**default IRD-schema-view.** The IRD-schema-view which automatically becomes the effective IRD-schema-view when an IRDS-user is recognized by the IRDS as ready to submit IRDS transactions.

**default IRD-view.** The IRD-view which automatically becomes the effective IRD-view when an IRDS-user is recognized by the IRDS as ready to submit IRDS transactions.

**derived data** - Data values which are derived from the values of other data by a specified algorithm.

**descriptive name.** In an IRDS, at IRD or IRD schema level, a unique and more descriptive name for an entity or meta-entity. It consists of the assigned descriptive name and the version identifier.

**distributed database.** A database that is physically located in two or more distinct locations.

**DML.** Data manipulation language.

**document administration.** The responsibility for management, control, formulation, use, and monitoring of documents throughout the system life cycle.

**documentation.** The aids provided for understanding the structure and intended uses of an information system or its components, e.g., flow charts, textual materials, end-user manuals, etc.

**domain.** The set of possible values of an attribute.

**E-R.** (1) Entity-relationship. (2) See also entity-relationship data model.

**edit.** To prepare data for a later operation. Editing may include the rearrangement or the addition of data, the deletion of unwarranted data, format control, code conversion, and the application of standard processes such as zero suppression.

**effective IRD-schema-view.** An IRD-schema-view which identifies the IRD Schema life-cycle-phase in which entities to be maintained exist. The effective IRDS-user can have only one effective IRD-schema-view at any one time. A user can change the effective IRD-schema-view with the set-session-defaults command.

**effective IRD-view.** An IRD-view which identifies the IRD-partition in which entities to be maintained exist. The effective IRDS-user can have only one effective IRD-view at any one time. A user can change the effective IRD-view with the set-session-defaults command.

**effective IRDS-user.** An IRDS-user entity which is associated by the IRDS with one or more IRDS transactions.

**entity.** In an IRDS, at IRD level, any person, place, thing, concept, or event about which data is or can be recorded.

**entity-relationship (E-R) model.** A data model based on the concept of entities, relationships among entities, and attributes of entities and relationships.

**entity-type.** In an IRDS, at IRD schema level, the label for a set of entities which have common attribute-types.

**exit.** Any instruction in a computer program, in a routine, or in a subroutine after the execution of which control is no longer exercised by that computer program, that routine, or that subroutine.

**export/import.** (1) In an IRDS, pertains to that set of commands, controls, and other procedural elements necessary to move the contents of one IRD to another. (2) See also portability.

**export/import procedure.** In an IRDS, the instructions to send to or accept from another IRDS all or a subset of its metadata.

**field.** In data processing, defined logical data that is part of a record.

**file.** A set of related records treated as a unit. A file may be computer oriented or may consist of a manual set of forms or records.

**functionality.** The capability to perform a function.

**hierarchical model.** A data model whose pattern of organization has the form of a tree structure.

**hierarchy.** (1) A set of directed relationships between two or more entities such that some entities are considered owners while others are members. This is distinguished from a network in that in a hierarchy, each member can have one and only one owner. (2) See also tree structure.

**implementation lock.** A lock imposed on meta-entities and meta-relationships by the implementor that prevents them from being deleted.

**information.** The meaning that a human being assigns to data by means of the conventions applied to that data.

**information resource.** The concept that the information resource which an enterprise uses for decision making and problem solving should be treated as a resource by the enterprise.

**information resource dictionary (IRD).** A collection of entities, relationships, and attributes used by an organization to model its information environment.

**information resource dictionary system (IRDS).** (1) A computer software system which provides facilities for recording, storing, and processing descriptions of an organization's significant information and information processing resources. (2) A computer software system which maintains and manages an information resource dictionary.

**information resource management.** The policy, action, or procedure concerning information (both automated and nonautomated) which management establishes that serve the overall current and future needs of the enterprise. Such policies, actions, or procedures would include considerations of availability, timeliness, accuracy, integrity, privacy, security, auditability, ownership, use, and cost-effectiveness.

**information retrieval.** Actions, methods, and procedures for recovering stored data to provide information on a given subject.

**integrity.** See data integrity, system integrity.

**interface.** A point of communication between two or more processes, persons, or other physical entities.

**interoperability.** (1) In data processing, a characteristic of software which allows it to be run on more than one type or size of computer and under more than one operating system. (2) See also portability.

**IRD.** The database of metadata. See also Information Resource Dictionary.

**IRD descriptor.** In an IRDS, at IRD level, any entity, relationship, relationship-class, attribute, or attribute-group.

**IRD life cycle phase.** In an IRDS, a phase in the life of an IRD entity, or set of IRD entities, used as a basis for a logical partition of the IRD.

**IRD life-cycle-phase integrity.** The rules which define what relationships can exist between entities in different IRD life-cycle-phases.

**IRD partition.** A logical subset of IRD entities. Each IRD entity exists in one and only one IRD partition. Every IRD life-cycle-phase is an IRD partition. There can exist other IRD partitions for special purposes. One such IRD partition is named SECURITY. All security entities are contained in this IRD partition. Each IRD partition is defined in the IRD Schema by a meta-entity of type IRD-PARTITION.

**IRD Schema.** A model of the logical structure of the IRD, consisting of components such as entity-types, relationship-types, and attribute-types.

**IRD Schema descriptor.** In an IRDS, at IRD schema level, any meta-entity, meta-relationship, meta-relationship-class, meta-attribute, or meta-attribute-group of the schema.

**IRD Schema extensibility.** (1) In an IRDS, the capability to add new IRD schema descriptors, i.e., entity-types, relationship-types, attribute-types, and attribute-group-types. (2) Contrast with IRDS extensibility.

**IRD Schema life-cycle-phase.** In an IRDS, a phase in the life of an IRD Schema meta-entity, or set of IRD Schema meta-entities, used as a basis for a logical partition of the IRD Schema.

**IRD Schema life cycle phase integrity.** The set of rules that restrict the meta-relationships that can be established between meta-entities and restrict the movement of meta-entities from one IRD Schema life cycle phase to another.

**IRD Schema view.** In an IRD, an entity of type SCHEMA-VIEW which enables an IRDS user to maintain IRD schema descriptors within a given schema life-cycle-phase.

**IRD structure.** The set of all entity types and relationship types that can be instantiated in the IRD. In Module 1, the IRD structure is defined by the contents of the CONTROLLED IRD Schema life cycle phase. In Module 4, the Extensible Life Cycle Phase Facility, the IRD structure is defined by the SCHEMA-STRUCTURE meta-entity named IRD-STRUCTURE.

**IRD view.** In an IRD, an entity of type IRD-VIEW which enables an IRDS user to maintain IRD descriptors within a given IRD partition.

**IRDS.** Information Resource Dictionary System.

**IRDS Database.** The collective name for the IRD and IRD Schema.



**IRDS extensibility.** (1) The capability to create new functionality in the IRDS. (2) Contrast with IRD schema extensibility.

**item.** An element of a set of data, e.g., a file may consist of a number of items such as records which in turn may consist of other items.

**language.** (1) A set of characters, conventions, and rules that is used for conveying information. (2) See also semantics, syntax.

**life-cycle.** A conceptual framework which is used to trace the evolution of objects of certain classes over time. A life-cycle is divided into phases. An object is identified as existing in a particular life-cycle-phase. An object shall exist in only one life-cycle-phase at given instance in time.

**life cycle phase.** In an IRDS, a phase in the life of an IRD entity, or set of IRD entities, used as a basis for a logical partition of the IRD.

**life cycle phase class.** Within the IRDS, a life cycle phase must be of one of three classes: UNCONTROLLED, CONTROLLED, and ARCHIVED. Entities or meta-entities in an ARCHIVED life cycle phase class are usually considered obsolete and no longer in use. Entities or meta-entities in a CONTROLLED life cycle phase class are usually considered as finished, tested, installed, and in active use. Entities or meta-entities in an UNCONTROLLED life cycle phase are considered planned, under development, and not yet in active use.

**logical data structure.** (1) An end-user view of the relationship between data elements. (2) See also physical data structure.

**meta-attribute.** In an IRDS, at IRD schema level, a property of a meta-entity or meta-relationship of its schema.

**meta-attribute-group.** In an IRDS, at IRD schema level, an ordered set of two or more meta-attributes used together.

**meta-attribute-group-type.** In an IRDS, at IRD schema description level, an ordered set of two or more meta-attribute-types used together, e.g., date and time-of-day.

**meta-attribute-type.** In an IRDS, at IRD schema description level, a property of a meta-entity-type or a meta-relationship-type.

**meta-entity.** (1) In an IRDS, at IRD schema level, a type instantiated in the IRD. (2) In an IRDS, at IRD schema level, a construct used for control.

**meta-entity-type.** In an IRDS, at IRD schema description level, the label for a set of meta-entities with common meta-attribute-types.

**meta-relationship.** In an IRDS, at IRD schema level, a directed association between meta-entities.

**meta-relationship-class.** In an IRD, at schema level, a label for a set of meta-relationships.

**meta-relationship-class-type.** In an IRD, at schema description level, a label for a set of meta-relationship types.

**meta-relationship-type.** In an IRDS, at IRD schema description level, the label for a set of meta-relationships with common meta-attribute-types.

**metadata.** Information about an organization's information and data activities and holdings.

**module.** (1) In a computer program, a self-contained subdivision that may be separately compiled. (2) In an IRDS, a set of capabilities that may be required or optional.

**network.** In CODASYL, a set of directed relationships between owner and member records such that a member record may belong to one or more owner relationships.

**network model.** A data model that consists of a modified tree structure that permits all but the root record to have multiple ancestor records.

**nonrepeating attribute.** An attribute such that a single entity or relationship can have at most one attribute of a given type.

**nonrepeating attribute-group.** An attribute-group such that a single entity or relationship can have at most one attribute-group of a given type.

**NUL.** The null character.

**null character (NUL).** A control character that is used to accomplish media-fill or time-fill and that may be inserted into or removed from a sequence of characters without affecting the meaning of the sequence; however, the control of equipment or the format may be affected by this character.

**panel interface.** In an IRDS, a screen-oriented user interface designed to permit interactive processing.

**physical data structure.** (1) The form in which data is stored on a medium. (2) See also logical data structure.

**portability.** (1) In data processing, the ability to transfer data from one system to another without being required to recreate or reenter data descriptions or to significantly modify the application being transported. (2) Synonymous with transportability. (3) See also interoperability.

**procedure.** The description of the course of action taken for the solution of a problem.

**program.** See computer program.

**QA.** Quality assurance.

**quality assurance.** All actions that are taken to ensure that a development organization delivers products that meet performance requirements and adhere to standards and procedures.

**record.** A set of related data or words treated as a unit, e.g., in stock control, each invoice could constitute one record.

**relationship.** In an IRDS, at IRD level, a directed association between entities.

**relationship-class.** In an IRDS, at IRD level, a label for a set of relationships.

**relationship-class-type.** In an IRDS, at IRD schema level, a label for a set of relationship-types.

**relationship-type.** In an IRDS, at IRD schema level, the label for a set of relationships which have common attribute-types.

**repeating attribute.** An attribute for which a single entity or relationship can have multiple (i.e.,  $n > 1$ ) attributes of that attribute-type.

**repeating attribute-group.** An attribute-group for which a single entity or relationship can have multiple (i.e.,  $n > 1$ ) attribute-groups of that attribute-group-type.

**Requirements Document.** (1) In TC X3H4, a collection of technical papers and proposals describing a set of requirements that an information resource dictionary system (IRDS) should satisfy. Requirements will include the capabilities, and entity, attribute and relationship types that the IRDS should contain. However, the concepts and capabilities described must represent state-of-the-art technical feasibility. Appearance in the Document does not imply inclusion in the Base Document, however, no requirement may appear in the Base Document unless it resides in the Requirements Document. (2) See also Base Document.

**retrieval.** (1) In information processing, the act or process of recovering information of data from storage. (2) See also information retrieval.

**retrieval command.** In the IRDS, a command in command language that includes data selection and which may be followed by some action on the selected data.

**revision number.** (1) In an IRDS, at IRD or IRD schema level, a positive integer consecutively assigned to each change affecting an IRD entity or meta-entity, respectively. (2) In an IRD, at data content or schema level, a component of the version identifier of an IRD entity or meta-entity, respectively. (3) See also version identifier.

**scan mask.** A string of characters used in object selection that defines a pattern which the name or string for which it is specified must match.

**security partition.** In an IRDS, a predefined IRD partition in which security-related entities must be stored and maintained.

**semantics.** (1) The relationships of characters or groups of characters to their meanings, independent of the manner of their use. (2) See also language, syntax.

**session.** In an IRDS, the period of time between initiation and completion of IRDS use.

**SLC.** System life cycle.

**syntax.** (1) The rules governing the structure of a language. (2) See also language, semantics.

**system.** (1) In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. (2) In data processing, a set of one or more computers, associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc., that form an autonomous whole capable of performing information processing and/or information transfer. (3) In information processing, the people, equipment, data, procedures, etc., which create, receive, enter, transmit, store, and otherwise use data within the enterprise to produce information necessary to meet desired objectives. May encompass any type of information processing system whether computerized or not.

**system integrity.** (1) In data processing, the state in which all system components (hardware, operating systems, programs, procedures, and data) are logically complete and correct, and protected. (2) See data integrity.

**system life cycle.** (1) Those phases and activities associated with, for example, the analysis, design, development, test, integration, operation, maintenance, and modification of a system. (2) See also life cycle phase.

**transportability.** Synonym for portability.

**tree structure.** In database systems, an arrangement of records that consists of a root, or base node, which serves as the ancestor of one or more descendent records of a hierarchy whose successively lower levels occur in a pattern that alternates the ancestor-descendant relationship. (2) See also hierarchy.

**user.** In information processing, an individual, organization or facility that makes use of an information processing system.

**user exit.** See exit.

**validation.** In data processing, the checking of data for correctness or compliance with applicable standards, rules, and conventions.

**validation procedure.** A set of rules and declarations, structured in order of precedence, which are used to evaluate the correctness of operations.

**variation name.** (1) In an IRDS, at IRD or IRD schema level, a label which identifies a grouping of entities or meta-entities used for control purposes. It is a component of the version identifier. The variation name of a meta-entity is always blank. (2) See also version identifier.

**verification.** The demonstration of consistency, completeness, and correctness of software at each stage and between each stage of the development life-cycle.

**version identifier.** In an IRDS, at IRD or IRD schema level, a label which distinguishes among entities or meta-entities with the same assigned access name. It consists of the variation name and the revision number and is a component of an access name or a descriptive name.

**view.** In an IRDS, a set of specified entity-types and relationship-types within a single life cycle phase.

## 4 Basic Elements

Within this section the basic terminal symbols and most common nonterminal symbols are defined. Basic grammatical rules are also identified.

### 4.1 Characters

**Function** Define the terminal symbols of the language and the elements of strings.

**Format**

```

digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

upper-case-letter ::=
    A | B | C | D | E | F | G | H | I | J | K | L | M |
    N | O | P | Q | R | S | T | U | V | W | X | Y | Z

lower-case-letter ::=
    a | b | c | d | e | f | g | h | i | j | k | l | m |
    n | o | p | q | r | s | t | u | v | w | x | y | z

space ::= /* blank character */

new-line ::= /* implementation-dependent end-of-line indicator */

vertical-bar ::= |

quote-character ::= "

special-character ::=
    + | - | * | / | = | $ | , | . | : | ; |
    ( | ) | & | ^ | % | # | @ | ! | > | < |
    ? | _ | { | } | [ | ] | \ | ~ | ' |
    vertical-bar | new-line | space | quote-character

null-mark ::= /* implementation-selected-special-character */

letter ::= upper-case-letter | lower-case-letter

character ::= letter | digit | special-character

letter-or-digit ::= letter | digit

```

**Syntax Rules**

- (1) Additionally, a special character shall be any character in the implementation-defined character set other than a digit or a letter.

**General Rules** None.

**4.2 Literals**

**Function** Specify a non-null value.

**Format**

```

asterisk ::= *

spacing-character ::= - | _

irds-name-special-character ::= & | % | # | $ | @

```

```

irds-name ::= naming-character { naming-character }

naming-character ::= spacing-character
                  | letter-or-digit
                  | irds-name-special-character

unsigned-integer ::= digit { digit }

signed-integer ::= [ + | - ] unsigned-integer

integer ::= signed-integer | unsigned-integer

fixed-point-literal ::= integer [.unsigned-integer]
                    | integer.
                    | [ + | - ] .unsigned-integer
/* fixed-point-literal is not used in Module 1 of the IRDS */

numeric-literal ::= integer-literal
                 | real-literal

date-time-literal ::= date-literal
                   | time-literal

integer-literal ::= integer

real-literal ::= signed-real | unsigned-real

signed-real ::= [+|-] unsigned-real

unsigned-real ::= non-fractional-real
              | fractional-real

non-fractional-real ::= unsigned-integer
                    exponent-character
                    scale-factor

fractional-real ::= unsigned-integer
                 decimal-point-character
                 fractional-part
                 [exponent-character
                 scale-factor]

exponent-character ::= e | E

scale-factor ::= signed-integer

decimal-point-character ::= .

fractional-part ::= unsigned-integer

date-literal ::= specified-date
              | current-date-word

```

```

specified-date ::= standard-format-date
                 | /* implementor-defined-format */

standard-format-date ::= [century-code]
                        year-code
                        month-code
                        day-code

current-date-word ::= CURRENT-DATE

time-literal ::= specified-time
               | current-time-word

specified-time ::= standard-format-time
                | /* implementor-defined-format */

standard-format-time ::= hour-code
                       minute-code
                       [second-code]

current-time-word ::= CURRENT-TIME

century-code ::= digit
              digit

year-code ::= digit
          digit

month-code ::= 01|02|03|04|05|06|07|08|09|10|11|12

day-code ::= 01|02|03|04|05|06|07|08|09|10|11|12
           |13|14|15|16|17|18|19|20|21|22|23|24
           |25|26|27|28|29|30|31

hour-code ::= 00|01|02|03|04|05|06|07|08|09|10|11
            |12|13|14|15|16|17|18|19|20|21|22|23

minute-code ::= 00|01|02|03|04|05|06|07|08|09|10
              |11|12|13|14|15|16|17|18|19|20|21
              |22|23|24|25|26|27|28|29|30|31|32
              |33|34|35|36|37|38|39|40|41|42|43
              |44|45|46|47|48|49|50|51|52|53|54
              |55|56|57|58|59

second-code ::= 00|01|02|03|04|05|06|07|08|09|10
              |11|12|13|14|15|16|17|18|19|20|21
              |22|23|24|25|26|27|28|29|30|31|32
              |33|34|35|36|37|38|39|40|41|42|43
              |44|45|46|47|48|49|50|51|52|53|54
              |55|56|57|58|59

string-literal ::= quote-character {character} quote-character

```

```

null-string ::= /* A quote-character immediately followed by the
                same quote-character */

long-string-literal ::= string-literal

short-string-literal ::= string-literal

literal ::= numeric-literal | long-string-literal
           | short-string-literal | irds-name

substitution-character ::= * | ?

scan-mask ::= name-scan-mask | number-scan-mask | string-scan-mask

number-scan-mask ::= numeric-literal | *

name-scan-mask ::=
    name-scan-mask-character {name-scan-mask-character}

name-scan-mask-character ::=
    spacing-character
    | irds-name-special-character
    | letter-or-digit
    | substitution-character

string-scan-mask ::=
    string-scan-mask-unit {string-scan-mask-unit}

string-scan-mask-unit ::=
    substitution-character
    | short-string-literal
    | irds-name-special-character }
    | spacing-character
    | letter-or-digit

```

### Syntax Rules

- (1) The same quote-character shall be used to delimit a string-literal.
- (2) If a quote-character shall be to be specified as a character within a string-literal, then two consecutive quote-characters shall be used to represent the single quote-character within the string.
- (3) No distinction shall be made between the null-string and a string-literal containing only blanks.
- (4) A long-string-literal can span multiple lines.
- (5) Substitution characters are used in selection criteria (refer to Subsection 6.33). Their meanings shall be as follows:
  - \* means any sequence of characters, including the null sequence.
  - ? means any single character other than null for an irds-name or string-literal.



- (6) All sequences of adjacent asterisks shall be equivalent to a single asterisk. (For example, \*\*\* means the same as \*.)
- (7) If two character sequences are separated by an asterisk in a single scan-mask, each defines a unique non-overlapping character sequence. (Thus, the following character sequences match the scan-mask ABC\*BCD:

ABCBCD          ABCABCD          ABCXYZBCD

However, ABCD does not match the same scan-mask).

- (8) All scan-masks shall be evaluated left to right, and the relative position of character sequences within a given scan-mask is significant. (Thus, the character sequence BCDABC would not match the scan-mask ABC\*BCD).
- (9) If any characters other than an irds-name-special-character or letter-or-digit is to be part of the scan-mask, they shall be enclosed in quote-characters. For example:

\*"?"\*          scans for a question-mark in the value

\*"A B"\*        scans for the character sequence: upper-case A, blank, upper-case B

- (10) Letters enclosed within quote-characters shall be case-sensitive in scanning. Letters not enclosed within quote-characters are not case-sensitive in scanning. In order to scan specifically for an upper-case-letter or lower-case-letter, the letter should be enclosed in quote-characters. Thus:

\*"Ob"\* matches "Object" but not "OBJECT", and

\*ob\* or \*OB\* matches "Object", "object" and "OBJECT".

- (11) In a scan-mask, the "-" spacing-character will match the "\_" spacing-character, and vice versa, unless the scan-mask is enclosed within quote-characters.

### General Rules

- (1) In an irds-name, a hyphen, "-", shall be equivalent to an underline, "\_".
- (2) In an irds-name, an upper-case-letter shall be equivalent to the corresponding lower-case-letter.
- (3) The use of plus-sign, minus-sign, decimal-point, and digit in a numeric-literal shall be in accordance with the conventions of fixed-point notation.
- (4) The maximum lengths of long-string-literal and short-string-literal shall be implementation-dependent. The maximum length of short-string-literal should be chosen so that a line of text can be entered without break via the primary input device in use. The maximum length of long-string-literal should enable an entire text attribute being specified in one long-string-literal. A minimum value for the maximum length of a long-string-literal shall be 32767. See Subsection 6.7.
- (5) The minimum range of integers shall be -32768 through 32767. The implementor may extend this range.

- (6) The maximum length of a line of text shall be implementation dependent. The minimum value for the maximum length of a line of text shall be 72 characters.
- (7) The use of CURRENT-DATE shall be the equivalent of the complete specification of the current date.
- (8) The use of CURRENT-TIME shall be the equivalent of the complete specification of the current date.
- (9) An invalid date shall not be specified.

### 4.3 Tokens

**Function** Specify lexical units.

**Format**

```

comment ::=
    slash asterisk {character}
    asterisk slash

token ::= regular-token | delimiter-token

regular-token ::= irds-name           4.2
                | scan-mask          4.2
                | irds-word
                | numeric-literal     4.2
                | date-time-literal
                | string-literal
                | comment

irds-word ::=

```

/\* All irds-words are listed alphabetically below. \*/  
/\* Those irds-words which are reserved are identified later. \*/

```

access-name-word ::= ACCESS-NAME
activate-word ::= ACTIVATE
add-word ::= ADD
after-word ::= AFTER
all-word ::= ALL
alternate-name-word ::= ALTERNATE-NAME
and-word ::= AND
ascending-word ::= ASCENDING
ASN-word ::= ASN
assigned-word ::= ASSIGNED
attribute-group-type-word ::= ATTRIBUTE-GROUP-TYPE
attributes-word ::= ATTRIBUTES
attribute-type-word ::= ATTRIBUTE-TYPE
attribute-type-validation-data-word ::= ATTRIBUTE-TYPE-VALIDATION-DATA
attribute-type-validation-procedure-word ::= ATTRIBUTE-TYPE-VALIDATION-
PROCEDURE
build-word ::= BUILD

```

check-word ::=	CHECK
command-imperative-word ::=	COMMAND-IMPERATIVE
context-word ::=	CONTEXT
controlled-word ::=	CONTROLLED
copy-word ::=	COPY
create-word ::=	CREATE
cumulative-word ::=	CUMULATIVE
current-word ::=	CURRENT
deactivate-word ::=	DEACTIVATE
decoded-word ::=	DECODED
defaults-word ::=	DEFAULTS
delete-word ::=	DELETE
descending-word ::=	DESCENDING
descriptive-name-word ::=	DESCRIPTIVE-NAME
difference-word ::=	DIFFERENCE
directly-word ::=	DIRECTLY
each-word ::=	EACH
encoded-word ::=	ENCODED
entities-word ::=	ENTITIES
entity-access-name-word ::=	ENTITY-ACCESS-NAME
entity-descriptive-name-word ::=	ENTITY-DESCRIPTIVE-NAME
entity-word ::=	ENTITY
entity-list-word ::=	ENTITY-LIST
entity-type-word ::=	ENTITY-TYPE
except-word ::=	EXCEPT
exclude-word ::=	EXCLUDE
exists-word ::=	EXISTS
exit-word ::=	EXIT
export-word ::=	EXPORT
file-word ::=	FILE
first-word ::=	FIRST
for-word ::=	FOR
format-word	FORMAT
forward-word ::=	FORWARD
from-word ::=	FROM
help-word ::=	HELP
highest-word ::=	HIGHEST
impact-word ::=	IMPACT
import-word ::=	IMPORT
in-word ::=	IN
increment-word ::=	INCREMENT
indirectly-word ::=	INDIRECTLY
individual-word ::=	INDIVIDUAL
intersect-word ::=	INTERSECT
inverse-word ::=	INVERSE
IRD-partition-word ::=	IRD-PARTITION
IRD-schema-word ::=	IRD-SCHEMA
IRD-schema-views-word ::=	IRD-SCHEMA-VIEWS
IRD-view-word ::=	IRD-VIEW
IRD-word ::=	IRD
IRDS-defaults-word ::=	IRDS-DEFAULTS
IRDS-limits-word ::=	IRDS-LIMITS
IRDS-reserved-names-word ::=	IRDS-RESERVED-NAMES
is-word ::=	IS

last-word ::=	LAST
life-cycle-phase-word ::=	LIFE-CYCLE-PHASE
lines-word ::=	LINES
list-name-word ::=	LIST-NAME
load-word ::=	LOAD
lowest-word ::=	LOWEST
maximum-word ::=	MAXIMUM
message-word ::=	MESSAGE
meta-attributes-word ::=	META-ATTRIBUTES
meta-entities-word ::=	META-ENTITIES
meta-entity-access-name-word ::=	META-ENTITY-ACCESS-NAME
meta-entity-descriptive-name-word ::=	META-ENTITY-DESCRIPTIVE-NAME
meta-entity-word ::=	META-ENTITY
meta-entity-type-word ::=	META-ENTITY-TYPE
meta-relationship-word ::=	META-RELATIONSHIP
meta-relationships-word ::=	META-RELATIONSHIPS
minimal-word ::=	MINIMAL
modify-word ::=	MODIFY
name-word ::=	NAME
names-word ::=	NAMES
new-word ::=	NEW
no-word ::=	NO
not-word ::=	NOT
of-word ::=	OF
on-word ::=	ON
only-word ::=	ONLY
or-word ::=	OR
order-word ::=	ORDER
other-word ::=	OTHER
output-word ::=	OUTPUT
page-word ::=	PAGE
panel-word ::=	PANEL
profiles-word ::=	PROFILES
quality-indicator-word ::=	QUALITY-INDICATOR
quality-word ::=	QUALITY
related-word ::=	RELATED
relationship-class-type-word ::=	RELATIONSHIP-CLASS-TYPE
relationship-word ::=	RELATIONSHIP
relationship-type-word ::=	RELATIONSHIP-TYPE
relationships-word ::=	RELATIONSHIPS
replace-word ::=	REPLACE
resequenced-word ::=	RESEQUENCED
revision-word ::=	REVISION
route-word ::=	ROUTE
save-word ::=	SAVE
select-word ::=	SELECT
sequence-word ::=	SEQUENCE
set-word ::=	SET
show-word ::=	SHOW
sort-word ::=	SORT
source-word ::=	SOURCE
start-word ::=	START
status-word ::=	STATUS
subtract-word ::=	SUBTRACT

```

suffix-word ::= SUFFIX
syntax-word ::= SYNTAX
target-word ::= TARGET
through-word ::= THROUGH
to-word ::= TO
union-word ::= UNION
unlock-word ::= UNLOCK
using-word ::= USING
variation-names-data-word ::= VARIATION-NAMES-DATA
variation-word ::= VARIATION
version-word ::= VERSION
view-word ::= VIEW
views-word ::= VIEWS
where-word ::= WHERE
with-word ::= WITH
xor-word ::= XOR

```

```

boolean-operator ::= and-word | or-word | xor-word | not-word

```

```

command-imperative ::=

```

/\* The command imperative shall be the sequence of irds-words which:

- (1) begin a command,
- (2) uniquely identify that command, and
- (3) are not part of any command-clause.

For example, the command-imperative for the add-entity-command consists of the two irds-words in order: add-word entity-word.

The concept is important for identifying a command and identifying where the next command shall be should the prior command be invalid and not properly terminated. \*/

```

delimiter-token ::= space | new-line
                  | relational-operator
                  | ( | ) | : | ; | , | '

```

```

relational-operator ::= equals
                    | greater-than
                    | less-than
                    | greater-than-or-equal
                    | less-than-or-equal
                    | not-equal

```

```

equals ::= =

```

```

greater-than ::= >

```

```

less-than ::= <

```

```

greater-than-or-equal ::= >=

```

```

less-than-or-equal ::= <=

```

not-equal ::=  $\diamond$

### Syntax Rules

- (1) An IRDS command stream consists of a series of regular tokens delimited by delimiter tokens
- (2) The implementor may provide abbreviations for each designated irds-word. The abbreviated form of an irds-word may be used wherever the full form can be used.
- (3) Boolean-operators have the following order of precedence:  
not-word, and-word, xor-word, or-word

**General Rules** None.

### 4.4 Entity and Meta-Entity Names

**Function** To identify rules for the identification of entities or meta-entities.

#### Format

```

IRD-name-space ::=
    { entity-access-name }
    { entity-descriptive-name }

entity-access-name ::=
    new-entity-access-name
    | existing-entity-access-name

new-entity-access-name ::= access-name

existing-entity-access-name ::= access-name

entity-descriptive-name ::= descriptive-name

IRD-schema-name-space ::=
    { meta-entity-access-name }
    { meta-entity-descriptive-name }

meta-entity-access-name ::=
    new-meta-entity-access-name
    | existing-meta-entity-access-name

new-meta-entity-access-name ::= access-name

existing-meta-entity-access-name ::= access-name

meta-entity-descriptive-name ::= descriptive-name

access-name ::=
    assigned-access-name
    [ version-identifier ]
    
```

```

descriptive-name ::=
    assigned-descriptive-name
    [ version-identifier ]

assigned-access-name ::= irds-name           4.2

assigned-descriptive-name ::= irds-name     4.2

version-identifier ::=
    ( variation-name : revision-number )
    | ( variation-name )
    | ( revision-number )

variation-name ::=
    letter           4.1
    [ irds-name ]   4.2
    | null          4.2

revision-number ::=
    unsigned-integer 4.2

name-scan-pattern ::=
    name-scan-mask   4.2
    [ ( name-scan-mask
    [ : number-scan-mask ] ) ] 4.2

```

**Syntax Rules**

- (1) The Syntax Rules for the IRD-name-space shall be as follows:
  - (a) The minimum and maximum length of an assigned-access-name for an entity can defined on an ENTITY-TYPE basis.
  - (b) The assigned-access-name for an entity shall conform to the format defined by the optional PICTURE meta-attributes of the entity's corresponding ENTITY-TYPE as defined in the IRD Schema. If no PICTURE meta-attributes are specified in the ENTITY-TYPE meta-entity, any irds-name shall be regarded as valid.
  - (c) Defaulting of version-identifier shall be as follows:
    - (i) If variation-name is not specified, a null variation-name shall be assumed.
    - (ii) If revision-number is not specified, then the highest existing revision-number with the specified or assumed variation-name shall be assumed.
- (2) The Syntax Rules for the IRD-schema-name-space shall be as follows:
  - (a) If revision-number is not specified, then the highest existing revision-number shall be assumed.
  - (b) For new-meta-entity-access-name:
    - (i) If version-identifier is specified, it shall be '(1)'.

- (ii) If version-identifier is not specified, then version-identifier defaults to '(1)'.
- (3) All characters, including spacing-characters, of an irds-name shall be significant and count in the total number of characters for the name.

### General Rules

- (1) The General Rules for the IRD-name-space shall be as follows:

#### For every entity

- (a) There shall be an entity-access-name which uniquely identifies the entity.
- (b) There can be a descriptive-name. If provided, the descriptive-name shall uniquely identify the entity.
- (c) Two entities can have the same assigned-descriptive-names, if and only if they have the same assigned-access-names.
- (d) The version-identifier of the entity's descriptive-name shall be the same as the version-identifier of its access-name.
- (e) The entity's assigned-access-name and assigned-descriptive-name can be identical.
- (f) The maximum allowed length of the entity's assigned-access-name shall be  $\geq 31$  characters.
- (g) The maximum allowed length of an assigned-descriptive-name shall not be less than the maximum length for the entity's assigned-access-name.
- (h) The maximum allowed length of an assigned-descriptive-name shall be the same as the maximum allowed length of a short-string-literal.
- (i) If  $n$  entities exist with the same assigned-access-name and the same revision-number, but with  $n$  different, non-null variation-names, then variation-name shall be used as part of the entity-access-name in order to uniquely identify each entity.
- (j) If the variation-name component of name-scan-pattern is an asterisk, the asterisk shall match a null variation-name.
- (k) For new-entity-access-name:
  - (i) The entity's assigned-access-name:
    - (A) shall not be the same as that of any entity which exists within the IRD, and
    - (B) cannot be the same as the assigned-descriptive-name of any entity which exists in the IRD.
  - (ii) If no variation-name is specified, a null value shall be assumed. A null variation-name shall always be valid.
- (l) For existing-entity-access-name:



- (i) The entity's assigned-access-name shall be the name of an entity which exists in the IRD.
- (ii) The variation-name shall default to null.
- (iii) The revision-number shall default to the highest revision-number of any entity with the same assigned-access-name and variation-name as existing-entity-access-name.
- (iv) The access-name shall identify an entity which is in the IRD-partition corresponding to the specified or default IRD-view(s) in effect.

**For every entity-type**

- (m) The absolute maximum length of the assigned-access-name for entities of that type shall be an implementor option.
- (n) Each system-generated assigned-access-name, for entities of that type, shall be of the form  $PN$ , where, if the meta-attribute of type START-NAME for that entity-type is  $Pk_1k_2 \dots k_j$ :
  - (i)  $P$  is the prefix specified for the START-NAME for the entity-type.
  - (ii)  $N$  is an integer that falls in the range  $L \leq N \leq U$ , where

$$L = (k_1) \cdot (10^{(k_j-1)}) + \dots + k_j, \text{ and}$$

$$U = (9) \cdot (10^{(k_j-1)}) + \dots + 9.$$

Each prefix shall be a valid irds-name.

- (o) The range of system-generated assigned-access-names shall be disjoint across all entity-types.
  - (p) Non-system-generated assigned-access-names (i.e., user-supplied assigned-access-names) shall never fall in the existing or potential range of any system-generated assigned-access-names.
- (2) The General Rules for the IRD-schema-name-space shall be as follows:

**For every meta-entity**

- (a) The revision-number shall have a value of 1 for a meta-entity of one of the following types:
  - IRD-PARTITION
  - IRDS-DEFAULTS
  - IRDS-LIMITS
  - IRDS-RESERVED-NAMES
  - NAMES
  - QUALITY-INDICATOR
- (b) There shall be a meta-entity-access-name which uniquely identifies the meta-entity.
- (c) There can be a meta-entity-descriptive-name. If provided, it shall uniquely identify the meta-entity.

- (d) The version-identifier of the meta-entity's meta-entity-descriptive-name shall be the same as the version-identifier of its meta-entity-access-name.
- (e) Two meta-entities can have the same assigned-descriptive-names if and only if they have the same assigned-access-names.
- (f) The meta-entity's assigned-access-name and assigned-descriptive-name can be identical.
- (g) For new-meta-entity-access-name, the meta-entity's assigned-access-name shall not be equal to:
  - (i) The assigned-access-name of any existing meta-entity,
  - (ii) The assigned-descriptive-name of any existing meta-entity,
  - (iii) The meta-entity-substitute-name of any existing meta-entity, or
  - (iv) The inverse-name of any existing meta-entity.
- (h) For existing-meta-entity-access-name:
  - (i) The specified assigned-access-name shall be the assigned-access-name of at least one meta-entity.
  - (ii) If version-identifier is specified, the resulting meta-entity-access-name shall exactly match the meta-entity-access-name of an existing meta-entity.
  - (iii) If version-identifier is not specified, the revision-number defaults to the highest revision-number for any existing meta-entity with the specified assigned-access-name.
  - (iv) The access-name shall identify a meta-entity which is in the IRD Schema life-cycle-phase corresponding to the specified or default IRD-schema-view(s) in effect.

There shall be no naming restrictions based on the meta-entity-type for meta-entity names.

### **Error and Warning Conditions**

- (1) Error E01007: Entity-access-name conflict. General Rule (1)(k) has been violated.
- (2) Error E01009: Entity does not exist or is not visible. General Rule (1)(l) has been violated.
- (3) Error E01065: Meta-entity already exists. General Rule (2)(g) has been violated.
- (4) Error E01066: Meta-entity assigned-access-name conflicts with meta-entity-substitute-name. General Rule (2)(g) has been violated.
- (5) Error E01068: Meta-entity assigned-access-name conflicts with inverse-name. General Rule (2)(g) has been violated.

- (6) Error E01067: Meta-entity does not exist or is not visible. General Rule (2)(h) has been violated.
- (7) Error E01069: Invalid meta-entity name format. One or more of the following situations has occurred:
- (a) The meta-entity's assigned-access-name part contained an invalid character.
  - (b) The meta-entity's version-identifier part is not valid.
  - (c) The length of the meta-entity's assigned-access-name exceeded the maximum length allowed.
- (8) Error E01070: Meta-entity assigned-access-name conflicts with assigned-descriptive-name of an existing meta-entity. General Rule (2)(g) has been violated.

#### 4.5 Attributes

**Function** Specify or identify the following:

- (1) The value of a non-repeating attribute;
- (2) A single value of a repeating attribute;
- (3) All values of a repeating attribute;
- (4) The value of a non-repeating attribute-group;
- (5) One attribute-group within a repeating attribute-group.

#### Format

*/\* the value of a non-repeating attribute or a single value of a repeating attribute \*/*

```

attribute ::=
    text-attribute | non-text-attribute

non-text-attribute ::=
    short-string-literal           4.2
    | irds-name                    4.2
    | numeric-literal              4.2
    | date-time-literal            4.2
    | null-mark                    4.1

text-attribute ::= string-literal-list

string-literal-list ::=           4.1
    string-literal [,string-literal-list] 4.2

```

*/\* all values of a repeating attribute \*/*

```
repeating-attribute-list ::=
    non-text-attribute
    [ , repeating-attribute-list ]
```

/\* the value of a non-repeating attribute-group or one attribute-group within a repeating attribute-group \*/

```
attribute-group ::=
    attribute-group-positional-format
    | attribute-group-non-positional-format
```

```
attribute-group-positional-format ::=
    ( attribute-1 | null
      { , attribute-i | null }
      [ , last-attribute-of-group ] )
```

```
attribute-group-non-positional-format ::=
    ( component-attribute-type-1-designator =
      component-attribute-1
      { [ , ] component-attribute-type-i-designator =
        component-attribute-i } )
```

```
attribute-1 ::= non-text-attribute
attribute-i ::= non-text-attribute
last-attribute-of-group ::= non-text-attribute
```

```
component-attribute-type-1-designator ::= irds-name 4.2
component-attribute-type-i-designator ::= irds-name 4.2
```

```
component-attribute-1 ::= non-text-attribute
component-attribute-i ::= non-text-attribute
```

### Syntax Rules

- (1) Each attribute shall conform to the format and value constraints of the associated ATTRIBUTE-TYPE as defined in the IRD Schema.
- (2) Each attribute of a repeating attribute shall be unique.
- (3) Within an attribute-group, null values shall not be allowed for component attributes whose corresponding attribute-types are SIGNIFICANT within the corresponding attribute-group-type. (A non-repeating attribute-group-type need not have any SIGNIFICANT component attribute-types. A repeating attribute-group-type shall have at least one SIGNIFICANT component attribute-type.)
- (4) In the attribute-group-non-positional-format, each component-attribute-type-designator shall be unique.
- (5) Any non-text-attribute can be enclosed in quotes, i.e., be specified as a short-string-literal. However, all validation rules for the attribute-type shall then apply to the contents of the short-string-literal. If null-mark is enclosed in quotes, the value of the attribute shall be assumed to be the null-mark character. (Note that it is possible for an ATTRIBUTE-TYPE to have an irds-word as a permissible value. Enclosing the value in quotes in this case will eliminate any ambiguity.)

- (6) The value of an attribute shall be enclosed in quotes if it contains any special-purpose character (as defined in Subsection 1.4) or any special-character which is not a naming-character.
- (7) No non-text-attribute can span a line, i.e., the new-line character shall not be valid in any non-text-attribute.
- (8) If multiple string literals are specified for a text attribute, each literal will define a set of one or more lines.

### General Rules

- (1) The order of attributes in a repeating-attribute-list shall not be significant.
- (2) Each attribute-*i* in an attribute-group-positional-format shall be associated with the component-attribute-type at the *i*-th relative position within the attribute-group-type as determined by the GROUP-POSITION meta-attribute of the ATTRIBUTE-GROUP-TYPE to ATTRIBUTE-TYPE meta-relationship-type.
- (3) Each component-attribute-type-*i*-designator in the attribute-group-non-positional-format shall be an ATTRIBUTE-TYPE in the IRD Schema.
- (4) Attribute-group shall be associated with an ATTRIBUTE-GROUP-TYPE defined in the IRD Schema.
- (5) Each component-attribute-type-*i*-designator specified in the attribute-group-non-positional-format shall be related to the corresponding ATTRIBUTE-GROUP-TYPE in the IRD Schema.
- (6) If an attribute-group-non-positional-format is used to declare a new or replacement value for an attribute-group or repeating attribute-group, and if a component ATTRIBUTE-TYPE of the ATTRIBUTE-GROUP-TYPE is not specified, then no change of value shall be assumed for the corresponding attribute.

### NOTES:

- (1) A repeating attribute is defined as one for which a single entity or relationship can have multiple (i.e.,  $n > 1$ ) attributes of that type. The same applies for a repeating attribute-group.
- (2) A non-repeating attribute is defined as one for which a single entity or relationship can have at most one attribute of that type. The same applies for a non-repeating attribute-group.

## 4.6 Type Designators

**Function** These are tokens used in the IRD language which refer to IRD Schema descriptors.

### Format

```

type-designator ::=
    entity-type-designator
    | relationship-type-designator
    | relationship-class-type-designator
    | attribute-type-designator

```

```

    | attribute-group-type-designator
entity-type-designator ::= meta-entity-designator
relationship-type-designator ::= meta-entity-designator
relationship-class-type-designator ::= meta-entity-designator
attribute-type-designator ::= meta-entity-designator
attribute-group-type-designator ::= meta-entity-designator
IRD-life-cycle-phase-designator ::= meta-entity-designator
quality-indicator-designator ::= meta-entity-designator

meta-entity-designator ::=
    meta-entity-assigned-access-name
    | meta-entity-substitute-name

meta-entity-assigned-access-name ::=
    assigned-access-name
    4.4

meta-entity-substitute-name ::= irds-name

path ::=
    forward-association-designator
    | inverse-association-designator

forward-association-designator ::=
    relationship-type-designator
    | relationship-class-type-designator

inverse-association-designator ::=
    relationship-type-inverse-name
    | relationship-class-type-inverse-name

relationship-type-inverse-name ::=
    inverse-name

relationship-class-type-inverse-name ::=
    inverse-name

inverse-name ::= irds-name

```

**Syntax Rules** None.

**General Rules**

- (1) Life-cycle-phase-designator shall be the assigned-access-name or meta-entity-substitute-name of a meta-entity of type IRD-PARTITION, with a LIFE-CYCLE-PHASE-CLASS meta-attribute.

- (2) Quality-indicator-designator shall be the assigned-access-name or meta-entity-substitute-name of a meta-entity of type QUALITY-INDICATOR.
- (3) Each type-designator shall be the assigned-access-name or meta-entity-substitute-name of a meta-entity in the CONTROLLED IRD Schema life-cycle-phase.
- (4) For type-designators:
  - (a) Entity-type-designator shall correspond to a meta-entity of type ENTITY-TYPE.
  - (b) Relationship-type-designator shall correspond to a meta-entity of type RELATIONSHIP-TYPE.
  - (c) Relationship-class-type-designator shall correspond to a meta-entity of type RELATIONSHIP-CLASS-TYPE.
  - (d) Attribute-type-designator shall correspond to a meta-entity of type ATTRIBUTE-TYPE.
  - (e) Attribute-group-type-designator shall correspond to a meta-entity of type ATTRIBUTE-GROUP-TYPE.
- (5) Inverse-association-designators shall be equal to an INVERSE-NAME meta-attribute of one meta-entity of type RELATIONSHIP-TYPE or RELATIONSHIP-CLASS-TYPE which is in the CONTROLLED IRD Schema life-cycle-phase.

#### 4.7 IRD Schema Language Tokens

**Function** To identify those tokens which are unique to the language and which interact with the IRD Schema.

#### Format

```

meta-attribute ::=
    short-string-literal           4.2
    | numeric-literal              4.2
    | irds-name                    4.2
    | null-mark                    4.1

repeating-meta-attribute-list ::=
    meta-attribute
    [ [,] repeating-meta-attribute-list ]

meta-attribute-group ::=
    ( meta-attribute-1 | null
      { , meta-attribute-i | null }
      [ , last-meta-attribute ] )

meta-attribute-group-list ::=
    meta-attribute-group
    [ [,] meta-attribute-group-list ]

meta-attribute-1 ::= meta-attribute
    
```

```

meta-attribute-i ::= meta-attribute

last-meta-attribute ::= meta-attribute

IRD-schema-life-cycle-phase ::=
    UNCONTROLLED
    | CONTROLLED
    | ARCHIVED

meta-relationship-identifier ::=
    meta-entity-1-access-name
    meta-relationship-type-or-class-type
    meta-entity-2-access-name
    [ key-meta-attribute ]

meta-entity-1-access-name ::=
    meta-entity-access-name                                4.4

meta-entity-2-access-name ::=
    meta-entity-access-name                                4.4

meta-relationship-type-or-class-type ::=
    meta-relationship-type
    | meta-relationship-class-type

meta-relationship-type ::=
    RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-CLASS-TYPE
    | RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE
    | RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE
    | RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE
    | ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE
    | ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE
    | ENTITY-TYPE-USES-VARIATION-NAMES-DATA
    | ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE
    | ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA
    | ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-PROCEDURE

meta-relationship-class-type ::=
    CONNECTS
    | CONTAINS
    | USES
    | MEMBER-OF

key-meta-attribute ::=
    simple-meta-attribute-command-clause                    6.106

meta-entity-type ::=
    entity-type-word                                        4.3
    | relationship-class-type-word                          4.3
    | relationship-type-word                                4.3
    | attribute-type-word                                   4.3
    | attribute-group-type-word                             4.3
    | IRD-partition-word                                    4.3
    | quality-indicator-word                                4.3

```



attribute-type-validation-procedure-word	4.3
attribute-type-validation-data-word	4.3
variation-names-data-word	4.3
names-word	4.3
IRDS-limits-word	4.3
IRDS-defaults-word	4.3
IRDS-reserved-names-word	4.3

meta-attribute-type ::=  
 non-repeating-meta-attribute-type  
 | repeating-meta-attribute-type  
 | component-meta-attribute-type  
 | text-meta-attribute-type

non-repeating-meta-attribute-type ::=  
 simple-meta-attribute-type  
 | system-maintained-meta-attribute-type

simple-meta-attribute-type ::=  
 ENTITY-CLASS  
 | FORMAT  
 | GROUP-POSITION  
 | INTEGER-LIMIT  
 | LINE-COUNT-LIMIT  
 | LINE-LENGTH-LIMIT  
 | MAXIMUM-ATTRIBUTE-LENGTH  
 | MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH  
 | MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT  
 | MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT  
 | MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH  
 | MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT  
 | MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT  
 | MAXIMUM-META-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT  
 | MAXIMUM-META-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT  
 | MAXIMUM-NUMBER-OF-OCCURRENCES  
 | MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT  
 | MAXIMUM-NUMBER-OF-OCCURRENCES-LIMIT  
 | MINIMUM-ATTRIBUTE-LENGTH  
 | MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH  
 | MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT  
 | MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH  
 | MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT  
 | ORIGIN  
 | STANDARD-MODE  
 | POSITION  
 | SEQUENCE-PARAMETER  
 | SEQUENCED  
 | SIGNIFICANT-ATTRIBUTES  
 | SINGULAR  
 | START-NAME  
 | STRING-LENGTH-LIMIT  
 | SYSTEM-GENERATED  
 | VALIDATION-TYPE  
 | VARIATION-LENGTH-LIMIT

system-maintained-meta-attribute-type ::=  
    audit-meta-attribute-type  
    | static-meta-attribute-type  
    | control-meta-attribute-type

audit-meta-attribute-type ::=  
    ADDED-BY  
    | LAST-MODIFIED-BY  
    | NUMBER-OF-TIMES-MODIFIED

static-meta-attribute-type ::=  
    COMMON-TO-ENTITY-TYPES  
    | CONNECTABLE  
    | TEXT-IN-GROUPS-ALLOWED

control-meta-attribute-type ::=  
    IMPLEMENTATION-LOCK  
    | LIFE-CYCLE-PHASE-CLASS  
    | NUMBER-OF-INSTANCES  
    | SYSTEM-LOCK  
    | SYSTEM-MAINTAINED

repeating-meta-attribute-type ::=  
    INVERSE-NAME  
    | META-ENTITY-SUBSTITUTE-NAME  
    | PICTURE  
    | RESERVED-ENTITY-NAME  
    | RESERVED-META-ENTITY-NAME  
    | VARIATION

component-meta-attribute-type ::=  
    DECODED-VALUE  
    | ENCODED-VALUE  
    | HIGH-VALUE  
    | LOW-VALUE  
    | SYSTEM-DATE  
    | SYSTEM-TIME

text-meta-attribute-type ::=  
    DESCRIPTION-OF-RULES  
    | PURPOSE

meta-attribute-group-type  
    repeating-meta-attribute-group-type  
    | audit-meta-attribute-group-type

repeating-meta-attribute-group-type ::=  
    DATA-RANGE  
    | DATA-VALUE

audit-meta-attribute-group-type ::=  
    DATE-TIME-ADDED  
    | DATE-TIME-LAST-MODIFIED

### Syntax Rules

- (1) Each meta-relationship has precisely one meta-relationship-type. The meta-relationship-types are defined in Subsection 9.2.
- (2) The meaning, purpose, and validation rules of each meta-attribute-type are defined in Subsection 9.3.
- (3) The association between meta-entity-types and meta-attribute-types is defined in Subsection 9.4.
- (4) The correspondence between meta-relationship-types and meta-attribute-types is defined in Subsection 9.5.
- (5) The association between meta-attribute-group-types and their components-meta-attribute-types is defined in Subsection 9.6.
- (6) The association between meta-entity-types and meta-attribute-group-types is defined in Subsection 9.7.
- (7) The implementor may provide abbreviations for each designated meta-relationship-class-type, meta-entity-type, meta-attribute-type, and meta-attribute-group-type.

### General Rules

- (1) Meta-entity-substitute-names for each meta-entity are META-ENTITY-SUBSTITUTE-NAME meta-attributes.
- (2) The inverses of relationship-type and relationship-class-type meta-entities shall be INVERSE-NAME meta-attributes of the corresponding meta-entities.
- (3) Simple-meta-attributes can have user designated values. They may be optional, required, or fixed within any meta-entity-type or meta-relationship-type. Consult Section 9 for details on specific meta-attribute-types.
- (4) System-maintained meta-attributes shall be maintained by the system. The user can test these values in selection criteria, and request that these meta-attributes be shown. The user shall not directly assign values to these meta-attributes.
- (5) Subsection 9.2.1 defines which meta-relationship-types have key-meta-attribute-types. A key-meta-attribute shall be specified only for meta-relationships of those types. The allowed values for such meta-attribute-types are defined in Subsection 9.3 in the description of the meta-attribute-types.
- (6) The rules for system-maintained meta-attributes are defined in the constraints for each such meta-attribute-type in Subsection 9.3.

### 4.8 Security Tokens

**Function** To identify those tokens which identify security descriptors.

**Format**

```
security-entity-name ::=
    assigned-access-name                                4.4
```

**Syntax Rules**

- (1) Version-identifiers shall not be specified for entities whose types are security-entity-types.

**General Rules**

- (1) There shall be one and only one security-partition-designator, named SECURITY.
- (2) All security entities shall be in the security-partition named SECURITY.
- (3) There shall be no relationship which relates any entity in the security-partition-designator named SECURITY to any entity in any other partition.
- (4) The security-attribute-types are those attribute-types in the CONTROLLED IRD Schema life-cycle-phase which are associated with either a security entity-type or a security relationship-type in the CONTROLLED IRD Schema life-cycle-phase.

**4.9 IRDS Functions**

**Function** To obtain some property of an IRD descriptor or IRD Schema descriptor. The value returned by the function can be used in selection criteria.

**Format**

```
irds-function ::=
    length-function
    | count-function
    | lines-function

length-function ::=
    length-function-qualifier'LENGTH

count-function ::=
    count-function-qualifier'COUNT

lines-function ::=
    lines-function-qualifier'LINES

length-function-qualifier ::=
    non-repeating-attribute-type-designator
    | non-repeating-meta-attribute-type           4.7
    | entity-access-name-word                   4.3
    | meta-entity-access-name-word             4.3
    | entity-descriptive-name-word             4.3
    | meta-entity-descriptive-name-word        4.3
```

count-function-qualifier ::=	
attribute-type-designator	4.6
meta-attribute-type	4.7
attribute-group-type-designator	4.6
meta-attribute-group-type	4.7
lines-function-qualifier ::=	
text-attribute-type-designator	
text-meta-attribute-type	4.7
non-repeating-attribute-type-designator ::=	
attribute-type-designator	4.6
text-attribute-type-designator ::=	
attribute-type-designator	4.6

### Syntax Rules

- (1) The function-identifier ('COUNT, 'LENGTH, 'LINES) shall be specified immediately after the corresponding function-qualifier. No intervening spaces shall be allowed.
- (2) Non-repeating-attribute-type-designator shall be a valid meta-entity assigned-access-name or meta-entity-substitute-name of an attribute-type meta-entity which is in the CONTROLLED life-cycle-phase, and for which the value of SINGULAR is YES within the qualifying entity-type or relationship-type.
- (3) Attribute-type-designator shall be a valid meta-entity-access-name or meta-entity-substitute-name of an attribute-type meta-entity in the CONTROLLED life-cycle-phase.
- (4) Attribute-group-type-designator shall be a valid meta-entity-access-name or meta-entity-substitute-name of an attribute-group-type meta-entity in the CONTROLLED life-cycle-phase.
- (5) Text-attribute-type-descriptor shall be a valid meta-entity-access-name or meta-entity-substitute-name of an attribute-type meta-entity which is in the CONTROLLED life-cycle-phase, and which has a PICTURE meta-attribute equal to TEXT.

### General Rules

- (1) The 'LENGTH function shall return an integer identifying the length in characters of the identified attribute or meta-attribute, or the length in characters of the assigned-access-name of the identified entity-access-name or meta-entity-access-name.
- (2) The 'COUNT function shall return an integer identifying the number of:
  - (a) Attributes or attribute-groups of the identified type, when an attribute-type or attribute-group-type has been specified. If the attribute or attribute-group does not exist within a given entity or relationship, the 'COUNT function shall return a value of zero. For existing non-repeating attributes and attribute-groups, the 'COUNT function shall return a value of 1.
  - (b) Meta-attributes or meta-attribute-groups of the identified type, when a meta-attribute-type or meta-attribute-group-type has been specified. If the meta-attribute or meta-attribute-group does not exist within a given meta-entity or meta-

relationship, the 'COUNT function shall return a value of zero. For existing non-repeating meta-attributes and meta-attribute-groups, the 'COUNT function shall return a value of 1.

- (3) The 'LINES function shall return an integer identifying the number of lines within the specified text attribute or text meta-attribute.

#### 4.10 The IRDS Data Model

NOTE: Within this discussion, the terms entity, relationship, attribute, and attribute-group are used in the context of defining a formal model.

The IRDS architecture is based upon two stores of information: the IRD and the IRD Schema.

The structure of both the IRD and the IRD Schema are defined in terms of an entity-relationship model.

This model is distinguished by the following characteristics:

- (1) It is a binary entity-relationship model. That is, a relationship is an association of two entities. Note that it is possible for an entity to be related to itself.
- (2) Relationships are directed associations. This means that a relationship from entity A to entity B is not the same as a relationship from entity B to entity A.
- (3) A relationship-class is used to denote a set of relationships. A relationship can belong to at most one relationship-class.
- (4) The model provides for attributes to be associated with either entities or relationships.
- (5) The model permits one level of grouping of attributes.
- (6) It is strongly typed. This means that each entity, relationship, attribute, and attribute-group has precisely one type. The type defines how instances are identified and what operations can be performed against those instances.

In particular:

- (a) For an entity, the entity-type defines how the entity can be named.
- (b) The relationship-type defines:
  - (i) whether or not a given set of entities can be associated
  - (ii) how the relationship can be specified
  - (iii) whether or not there can exist more than one relationship of a given type associating the same two entities.
- (c) For an attribute, the attribute-type defines how the attribute is identified and what maintenance operations can be performed against it.

- (d) For an attribute-group, the attribute-group-type defines:
- (i) which types of attributes can be grouped,
  - (ii) the ordering of attributes within the group,
  - (iii) which attributes shall be specified whenever there are multiple instantiations of the attribute-group-type within a given entity or relationship.

Any given entity, relationship, relationship-class, attribute, or attribute-group is called a descriptor. In order to distinguish between those descriptors within the IRD and the IRD Schema, the following terminology is used:

The term IRD descriptor refers to any descriptor which resides in the IRD. The term IRD Schema descriptor refers to any descriptor which resides in the IRD Schema.

Except for the above initial discussion, the terms entity, relationship, relationship-class, attribute and attribute-group are IRD descriptors.

In order to distinguish IRD descriptors from IRD Schema descriptors, the prefix "meta-" is used in reference to the IRD Schema. Thus the terms meta-entity, meta-relationship, meta-relationship-class, meta-attribute and meta-attribute-group are IRD Schema descriptors.

For clarity, the following is defined:

- The type of an entity is an entity-type.
- The type of a relationship-class is a relationship-class-type.
- The type of a relationship is a relationship-type.
- The type of an attribute is an attribute-type.
- The type of an attribute-group is an attribute-group-type.
- The type of a meta-entity is a meta-entity-type.
- The type of a meta-relationship-class is a meta-relationship-class-type.
- The type of a meta-relationship is a meta-relationship-type.
- The type of a meta-attribute is a meta-attribute-type.
- The type of a meta-attribute-group is a meta-attribute-group-type.

Note that the association between the IRD and IRD Schema can now be stated as follows:

Every entity-type, relationship-type, attribute-type, and attribute-group-type is represented by a meta-entity.

There are significant differences between the IRD Schema and the IRD:

- (1) Since the types of entities, relationships, attributes, and attribute groups which can reside in the IRD are entities within the IRD Schema, this set of types can be modified by an installation using IRD Schema maintenance commands.
- (2) Unlike the IRD, the types of entities, relationships, attributes, and attribute-groups which can exist in the IRD Schema are predetermined by the IRDS standard. Any changes to these result only from modifications in the maintenance and further enhancement of the standard.

- (3) In the IRD Schema, a relationship-type is defined completely by a pair of entity-types. In the IRD, there can be multiple relationship-types involving the same pair of entity-types.
- (4) For the IRD, all relationship-types shall be unconstrained; i.e., the ratio (as defined in Subsection 10.2.2) of all relationship-types is (0,n:0,m). In the IRD Schema, a meta-relationship-type has the ratio (0,n:0,1), (0,n:2), or (0,n:0,m).

#### 4.11 The IRD Schema Definition

Figure 1 shows the meta-entity-types and meta-relationship-types which exist in the IRD Schema. The ratio of each meta-relationship-type is also shown in this figure.

Details of these meta-entity-types and meta-relationship-types, as well as the meta-attribute-types and meta-attribute-group-types associated with these, are given in Section 9.

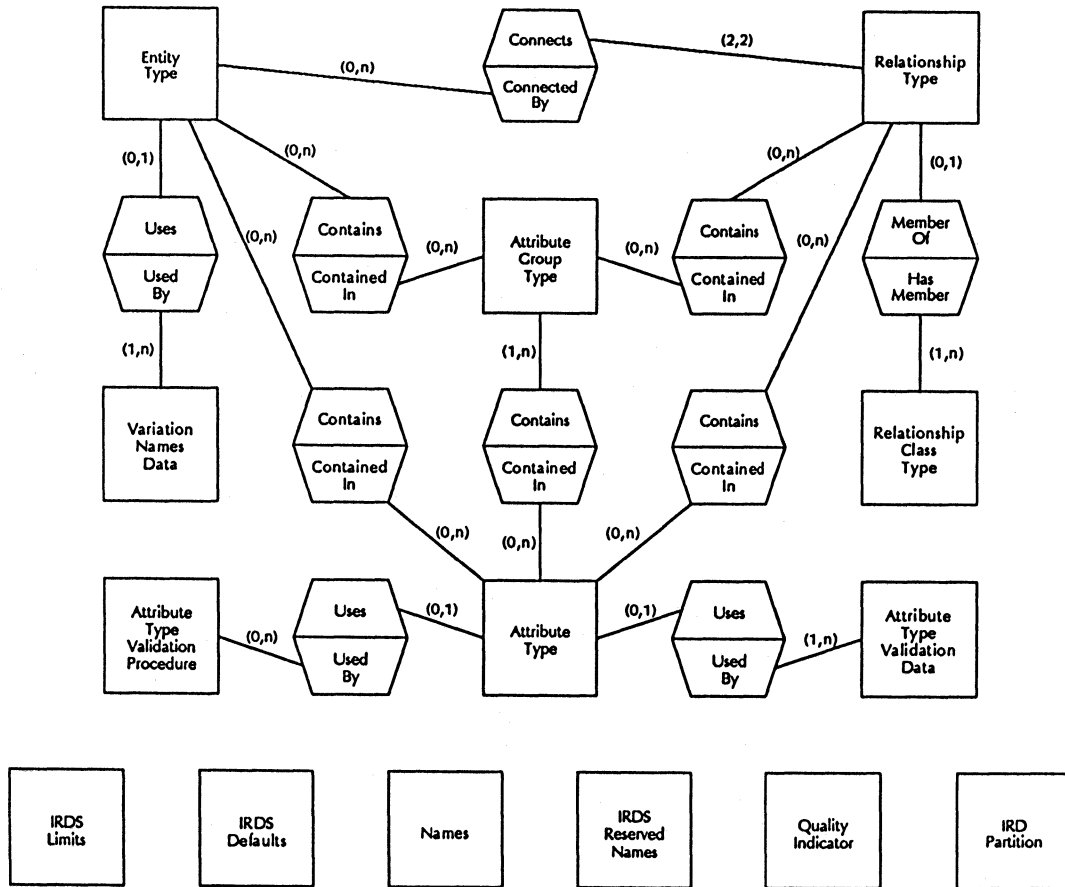
#### 4.12 Reserved Words

**Function** Specify reserved words.

**Format**

irds-module-1-reserved-word :=	
access-name-word	4.2
all-word	4.2
alternate-name-word	4.2
and-word	4.2
attributes-word	4.2
descriptive-name-word	4.2
entities-word	4.2
entity-type-word	4.2
exclude-word	4.2
first-word	4.2
for-word	4.2
increment-word	4.2
life-cycle-phase-word	4.2
lines-word	4.2
meta-entities-word	4.2
meta-entity-type-word	4.2
meta-relationships-word	4.2
new-word	4.2
not-word	4.2
of-word	4.2
only-word	4.2
or-word	4.2
order-word	4.2
quality-word	4.2
relationship-type-word	4.2
relationships-word	4.2
resequenced-word	4.2
revision-word	4.2
start-word	4.2
select-word	4.2





**Figure 1**  
**Meta-Entity-Types and Meta-Relationship-Types in the IRD Schema**

to-word	4.2
variation-word	4.2
version-word	4.2
with-word	4.2
xor-word	4.2

### Syntax Rules

- (1) An assigned-access-name shall not equal a reserved-word.
- (2) An assigned-descriptive-name shall not equal a reserved-word.
- (3) An attribute which equals a reserved word shall be enclosed in quotes.
- (4) An implementation may extend the list of reserved words.

### General Rules     None

NOTE: Syntax Rule (4) is required because of the potential of additional Modules of the standard and implementation extensions.

## 5 Command Specifications

**Function**     To identify major classes of commands and describe rules which apply to all commands.

### Format

/\* None. This breakdown is explanatory only. \*/

```
irds-command ::=
    IRD-schema-command           5.1
    | IRD-command                 5.2
    | general-command             5.3
```

### Syntax Rules

- (1) All commands shall begin with a valid command imperative, i.e., a sequence of irds-words which identify the command. The command imperative is then optionally followed by a list of IRD or IRD Schema objects which the command is to operate against, and/or one or more command-clauses.
- (2) Every command shall be terminated by a special character called the command-terminator. The semi-colon shall be the command terminator. In batch mode, if end of input is reached without valid command termination, a warning condition is raised.

### General Rules

- (1) Command specifications are numbered hierarchically. The number of periods in this scheme indicate the depth of the Section in the hierarchy. The more periods, the greater is the depth of a Section in the hierarchy. A common sequence of leading numbers separated by periods indicate a common "branch" of the tree. Sections identified by leading numbers are regarded as superior to other Sections lower on the same "branch." Thus Section "5" is superior to Subsection "5.x"; "5.x" is superior to "5.x.y", and "5.x.y" and "5.x.z" are at the same level: both are subordinate to "5.x".

The hierarchy of command specifications is used to identify common rules and/or actions. A rule or action specified at any given Section applies to all subordinate Sections unless expressly overridden.

- (2) Any command which has a syntax error will not execute.
- (3) A command which operates against multiple IRD Schema or IRD objects shall process as if it were multiple instances of the same command operating against individual IRD or IRD Schema objects.
- (4) Multiple IRD Schema or IRD objects can be specified directly within the command. Additionally, IRD commands support the technique for indirectly specifying IRD objects through the use of a selection criteria command-clause.
- (5) If multiple objects are directly specified, validation specified by general rules and security rules shall be performed on an object by object basis. If any rule is violated for a single specified object, the command proceeds to the next specified object.
- (6) If a selection criteria command-clause is specified, validation specified by general rules and security rules shall be performed on each entity or relationship obtained in accordance with the selection criteria. If an error occurs in processing any entity or relationship so obtained, all appropriate error and warning messages shall be produced, and the command shall continue processing. If no entities or relationships are obtained by executing the selection criteria, a warning message shall be produced.
- (7) The term, *user*, means the individual who requests the IRDS to perform one of the functions specified by the commands specified within this Section. Each user is presumed to have a corresponding entity of type IRDS-USER defined in the IRD. This entity is called the effective IRDS-user.

**Security Rules** None.

#### **Actions**

- (1) If a valid command imperative is not recognized, input shall be bypassed until a command terminator is encountered or a valid command imperative is recognized. The system shall indicate what input was bypassed.
- (2) If a valid command imperative is encountered while bypassing input in search of a command terminator, a warning message shall be produced. Normal command processing shall be resumed with the command imperative. The command is then processed according to the rules specified above.
- (3) If no command terminator or valid command imperative is recognized, the system shall indicate that no valid input was recognized.
- (4) A warning message shall be produced at the end of a command which encounters errors while processing multiple entities.

#### **Error and Warning Conditions**

- (1) Error E01001: Unrecognized or invalid command format. See Syntax Rule (1).

- (2) Warning W01001: End of input encountered and last command not properly terminated. See Syntax Rule (2).
- (3) Warning W01012: No entities selected. See General Rule (7).
- (4) Error E01016: Specified entity-list does not exist. See General Rule (8)(a).
- (5) Warning W01003: Specified entity-list is empty. See General Rule (8)(b).
- (6) Warning W01004: Command executed with errors. See Action (4).

NOTES:

- (1) The concept of effective IRDS-user applies to both single-user and multi-user implementations of the IRDS. This Standard does not presume either type of implementation.
- (2) For multi-user implementations, it is presumed that different users will be distinguished by different effective IRDS-users. In these cases, it is also assumed that the implementation will provide a mechanism for user identification and verification. This standard makes no constraints on such mechanisms.

### 5.1 IRD Schema Commands

**Function** To identify classes of commands which operate against the IRD Schema, and the rules which apply to all commands in these classes.

**Format**

/\* None. This breakdown is explanatory only. \*/

```
IRD-schema-command ::=
    IRD-schema-maintenance-command      5.1.1
  | IRD-schema-output-command          5.1.2
```

**Syntax Rules** None.

**General Rules**

- (1) The effective IRDS-user shall have at least one associated IRD-SCHEMA-VIEW entity. This is defined in the IRD by an entity of type IRD-SCHEMA-VIEW which is related to the effective IRDS user by a relationship of type IRDS-USER-HAS-IRD-SCHEMA-VIEW.
- (2) An IRD-SCHEMA-VIEW enables the effective IRDS-user to access and update meta-entities in a given IRD Schema life-cycle-phase. This correspondence between an IRD Schema View and an IRD Schema life-cycle-phase is defined by the attribute IRD-SCHEMA-PHASE-NAME. For any entity of type IRD-SCHEMA-VIEW, this attribute shall exist and be equal to one of the following values: UNCONTROLLED, CONTROLLED, ARCHIVED.
- (3) The effective IRDS user can have multiple IRD-SCHEMA-VIEWS. For updating the IRD Schema, only one of the IRD-SCHEMA-VIEW entities shall be in effect at any one time. This is called the effective IRD Schema view. A default effective IRD Schema view is determined by the relationship of type IRDS-USER-HAS-IRD-SCHEMA-VIEW. For any particular IRDS-USER entity, one and only one relationship of this type shall have the

DEFAULT-VIEW attribute equal to YES. The IRD-SCHEMA-VIEW entity in this relationship is the default effective IRD Schema view.

- (4) The user can change the effective IRD Schema view by using the set-session-defaults command (Subsection 5.3.2.1).

**Security Rules** None.

**Actions**

- (1) Upon entering the IRDS, the effective IRDS-user shall be assigned a default IRD-SCHEMA-VIEW, provided one has been defined.

**Error and Warning Conditions**

- (1) Error E01062: User has no IRD-schema-view. Command cannot be executed.

**5.1.1 IRD Schema Maintenance Commands**

**Function** To identify those commands which update the IRD Schema and the rules which apply to these commands.

**Format**

/\* None. This breakdown is explanatory only. \*/

IRD-schema-maintenance-command ::=	
add-meta-entity-command	5.1.1.1
modify-meta-entity-command	5.1.1.2
delete-meta-entity-command	5.1.1.3
add-meta-relationship-command	5.1.1.4
modify-meta-relationship-command	5.1.1.5
delete-meta-relationship-command	5.1.1.6
modify-meta-entity-access-name-command	5.1.1.7
modify-meta-entity-descriptive-name-command	5.1.1.8
modify-meta-entity-life-cycle-phase-command	5.1.1.9
copy-meta-entity-command	5.1.1.10
deactivate-IRD-command	5.1.1.11
restore-IRD-schema-command	5.1.1.12
activate-IRD-command	5.1.1.13

**Syntax Rules** None.

**General Rules** None.

**Access Rules**

For the following Access Rules:

- Let *U* denote the effective IRDS-user.
- Let *V* denote the effective IRD-schema-view.
- Let *V1* ... *Vn* denote IRD-schema-views which are related to *U*.

- (1) For add-meta-entity-command, copy-meta-entity command, and modify-meta-entity-command with new-meta-entity-version-clause specified, *V* shall have the attribute IRD-SCHEMA-PHASE-NAME equal to UNCONTROLLED provided the modify-meta-entity-life-cycle-phase command applies for the meta-entity-type of the meta-entity being maintained. If the meta-entity is of a type for which modify-meta-entity-life-cycle-phase does not apply, then *V* shall have the attribute IRD-SCHEMA-PHASE-NAME equal to CONTROLLED for the add-meta-entity and copy-meta-entity-command. (The new-meta-entity-version command-clause is not valid for meta-entities which cannot be moved from one IRD Schema life-cycle-phase to another.)
- (2) For the delete-meta-entity-command and the modify-meta-entity-command without a new-meta-entity-version-clause specified, *V* shall have an IRD-SCHEMA-PHASE-NAME attribute equal to the IRD Schema life-cycle-phase which contains the specified meta-entity.
- (3) For the meta-relationship maintenance commands, add-meta-relationship, modify-meta-relationship, and delete-meta-relationship:
  - (a) *V* shall have an IRD-SCHEMA-PHASE-NAME attribute equal to the IRD Schema life-cycle-phase which contains the first meta-entity of the specified meta-relationship.
  - (b) There shall be a  $V_i$ ,  $1 \leq i \leq n$ , with an IRD-SCHEMA-PHASE-NAME attribute equal to the IRD Schema life-cycle-phase which contains the second meta-entity of the meta-relationship.
- (4) For modify-meta-entity-life-cycle-phase-command:
  - (a) *V* shall have an IRD-SCHEMA-PHASE-NAME attribute equal to the new IRD Schema life-cycle-phase.
  - (b) There shall be a  $V_i$ ,  $1 \leq i \leq n$ , with an IRD-SCHEMA-PHASE-NAME attribute equal to the current IRD Schema life-cycle-phase, (which contains the meta-entities specified).

**Security Rules** None.

**Actions** None.

#### **Error and Warning Conditions**

- (1) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See Access Rules (1), (2), (3)(a), and (4)(a).
- (2) Error E01067: Meta-entity does not exist or is not visible. See Access Rules (3) and (4).

#### **5.1.1.1 Add Meta-Entity Command**

**Function** To add a meta-entity to the IRD Schema.

#### **Format**

```
add-meta-entity-command ::=
    add-word                4.3
    meta-entity-word        4.3
    new-meta-entity-access-name 4.4
```

meta-entity-type-clause	6.101
[ meta-entity-descriptive-name-clause ]	6.126
[ new-meta-attributes-clause ]	6.103
;	

**Syntax Rules** None.

### General Rules

- (1) The meta-entity-type specified in meta-entity-type-clause shall not be one of the following: IRDS-LIMITS, IRDS-DEFAULTS, RESERVED-NAMES, NAMES, and ATTRIBUTE-TYPE-VALIDATION-PROCEDURE. All meta-entities of one of these types shall be created by the IRDS when the IRD Schema is created.
- (2) If the meta-entity-type specified in meta-entity-type-clause is IRD-PARTITION or QUALITY-INDICATOR, then the effective IRD-schema-view shall have CONTROLLED as the value of the attribute-type IRD-SCHEMA-PHASE-NAME.
- (3) If meta-entity-type-clause identifies a valid meta-entity-type other than one of those identified in General Rules (1) and (2), then the effective IRD-schema-view shall have UNCONTROLLED as the value of the attribute-type IRD-SCHEMA-PHASE-NAME.
- (4) No meta-entity shall be created in the ARCHIVED IRD-schema-life-cycle-phase.

**Security Rules** None.

### Actions

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) If no errors were encountered in validation, then:
  - (a) The meta-entity shall be added to the IRD Schema in the IRD Schema life-cycle-phase identified by the IRD-SCHEMA-PHASE-NAME attribute in the effective IRD-schema-view.

The audit meta-attribute-group DATE-TIME-ADDED shall be created. Within this meta-attribute-group:

```
SYSTEM-DATE = date of transaction
SYSTEM-TIME = time of transaction.
```

The following meta-attributes shall also be created with the following values:

```
ADDED-BY = assigned-access-name of the effective IRDS-user
NUMBER-OF-TIMES-MODIFIED = 0
NUMBER-OF-INSTANCES = 0 (if applicable)
```

- (b) If meta-entity-descriptive-name-clause is specified, the meta-entity shall be given the descriptive-name with the assigned-descriptive-name specified in the command-clause.
- (c) All meta-attributes specified in new-meta-attributes-clause shall be created for the specified meta-entity.

- (d) If the meta-entity-type specified has any required meta-attribute-types which are not specified in new-meta-attributes-clause, the corresponding meta-attributes shall be created for the meta-entity with the default value. (See Subsection 9.4.)
- (e) If the specified meta-entity-type is ENTITY-TYPE, then meta-relationships shall be established from the specified meta-entity to all ATTRIBUTE-TYPE and ATTRIBUTE-GROUP-TYPE meta-entities with a meta-attribute of COMMON-TO-ENTITY-TYPES equal to YES.

If there do not exist two meta-entities of type ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE with the same assigned access-name, then:

- (i) If the specified ENTITY-TYPE and the ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE are both in the UNCONTROLLED IRD Schema life-cycle-phase, the meta-relationships are established between these;
- (ii) If the specified ENTITY-TYPE is in the UNCONTROLLED IRD Schema life-cycle-phase and the ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE is in the UNCONTROLLED IRD Schema life-cycle-phase, the meta-relationships are established between these.

If there do exist two meta-entities of type ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE with the same assigned access-name (one in the CONTROLLED and one in the UNCONTROLLED IRD Schema life-cycle-phase), then:

- (i) If the specified ENTITY-TYPE is in the CONTROLLED IRD Schema life-cycle-phase meta-relationships, are established between the ENTITY-TYPE and the ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE is in the CONTROLLED IRD Schema life-cycle-phase.
- (ii) If the specified ENTITY-TYPE is in the UNCONTROLLED IRD Schema life-cycle-phase, meta-relationships are established between the ENTITY-TYPE and the ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE is in the UNCONTROLLED IRD Schema life-cycle-phase.

- (3) Completion of command processing shall be confirmed.

### **Error and Warning Conditions**

- (1) Error E01071: Meta-entities of specified type are system created. See General Rule (1).
- (2) Error E01072: Meta-entities of specified type shall be created in CONTROLLED IRD Schema life-cycle-phase. See General Rule (2).
- (3) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See General Rules (2), (3), and (4).

### **5.1.1.2 Modify Meta-Entity Command**

**Function** To modify the meta-attributes of an existing meta-entity.



**Format**

```

modify-meta-entity-command ::=
    modify-word                4.3
    meta-entity-word          4.3
    existing-meta-entity-access-name 4.4
    [ meta-entity-type-clause ]    6.101
    [ new-meta-entity-version-clause ] 6.127
    [ meta-entity-descriptive-name-clause ] 6.126
    modified-meta-attributes-clause 6.104
    ;

```

**Syntax Rules** None.**General Rules**

For all the following General Rules:

Let *E* denote the meta-entity identified by existing meta-entity-access-name.  
 Let *T* be the type of *E*.  
 Let *P* be the IRD-schema-life-cycle-phase which contains *E*.  
 Let *U* denote the effective IRDS user.  
 Let *V* denote the effective IRD-schema-view.  
 Let *P'* denote the IRD Schema life-cycle-phase identified by the IRD-SCHEMA-PHASE-NAME attribute of *V*.

- (1) If meta-entity-type-clause is specified, the meta-entity-type specified in this command-clause shall be *T*.
- (2) Meta-entity-descriptive-name-clause shall be allowed only if *E* has no descriptive-name.
- (3) If *T* is a type for which modify-meta-entity-life-cycle-phase is allowed, then:
  - (a) If *P* is the CONTROLLED or ARCHIVED IRD Schema life-cycle-phase, then new-meta-entity-version-clause shall be required. (Equivalently, a meta-entity which can be moved between IRD Schema life-cycle-phases shall not be modified in the CONTROLLED or ARCHIVED phase.)
  - (b) If *P* is the UNCONTROLLED IRD Schema life-cycle-phase, new-meta-entity-version-clause is optional.
- (4) If *T* is a type for which the modify-meta-entity-life-cycle-phase command is not allowed, then new-meta-entity-version-clause shall not be allowed. (In this case, *T* shall be in the CONTROLLED IRD Schema life-cycle-phase.)
 

/\* This means that versioning does not apply to meta-entities which cannot move between life-cycle-phases, and that such meta-entities can only be modified in the CONTROLLED IRD Schema life-cycle-phase. \*/
- (5) If new-meta-entity-version-clause is not specified, *P'* shall equal *P*.
- (6) If new-meta-entity-version-clause is specified, then:
  - (a) *P'* shall be UNCONTROLLED.

- (b) There shall exist an IRD-schema-view related to *U* with an IRD-SCHEMA-PHASE-NAME attribute equal to *P*.
  - (c) For each meta-relationship in which *E* is the first meta-entity, a corresponding meta-relationship with the new meta-entity as the first meta-entity, the same second meta-entity, and the same meta-attributes shall be created, unless the corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity.
- (7) The command shall not operate on the following meta-entities:

DATE-TIME-ADDED  
DATE-TIME-MODIFIED  
ADDED-BY  
MODIFIED-BY  
NUMBER-OF-TIMES-MODIFIED  
SYSTEM-DATE  
SYSTEM-TIME

**Security Rules**     None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.

If the meta-entity has a non-zero value for the NUMBER-OF-INSTANCES meta-attribute, the contents of the IRD shall be checked to determine if the resulting modifications would invalidate any of the present IRD contents.

If any such conflicts arise, diagnostics shall be produced. The diagnostics shall clearly identify the nature of the conflict.

- (2) If no errors were encountered in validation and new-meta-entity-version-clause was not specified, then
- (a) The meta-attributes shall be modified as specified.
  - (b) Appropriate values shall be assigned to the following meta-attributes for the meta-entity:

NUMBER-OF-TIMES-MODIFIED = prior value + 1  
LAST-MODIFIED-BY = the assigned-access-name of the effective IRDS-user

- (c) The meta-attribute-group DATE-TIME-MODIFIED shall be updated as follows:

SYSTEM-DATE = date of transaction  
SYSTEM-TIME = time of transaction

- (d) If meta-entity-descriptive-name-clause is specified, then each meta-entity with the specified assigned-access-name shall be given the specified assigned-descriptive-name. The audit meta-attributes for each such meta-entity shall be changed as in (b) and (c) above.

- (3) If no errors were encountered and new-meta-entity-version-clause was specified, then
- (a) A new meta-entity of the same type as the existing meta-entity shall be created.
    - (i) The new meta-entity has the same assigned-access-name and same assigned descriptive-name as the existing meta-entity.
    - (ii) The new meta-entity's audit meta-attributes shall be as specified in the add-meta-entity-command.
    - (iii) For each meta-attribute specified in modified-meta-entity-attributes-clause, the new meta-entity has the resulting meta-attribute as if the corresponding meta-attribute in the existing meta-entity were modified as specified.
    - (iv) For each user-modifiable meta-attribute in the existing meta-entity which is not specified in modified-meta-attributes-clause, the new meta-entity shall be given an identical meta-attribute.
  - (b) If the existing meta-entity does not have a descriptive-name and meta-entity-descriptive-name-clause is specified, then each meta-entity with the specified assigned-access-name shall be given the specified assigned-descriptive-name. The audit meta-attributes for each such meta-entity shall be changed as in (3)(b) and (3)(c).
  - (c) Corresponding meta-relationships shall be created as specified in General Rule (6)(c). For each corresponding meta-relationship which cannot be created, a warning message shall be produced.
- (4) Completion of command processing shall be confirmed.

#### **Error and Warning Conditions**

- (1) Error E01074: Meta-entity is not of specified type. See General Rule (1).
- (2) Error E01075: Meta-entity already has a descriptive-name. See General Rule (2).
- (3) Error E01076: Meta-entity cannot be modified. New version is required. See General Rule (3)(a).
- (4) Error E01077: New version is not allowed for specified meta-entity. See General Rule (4).
- (5) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See General Rules (5) and (6).
- (6) Error E01067: Meta-entity does not exist or is not visible. See General Rule (6)(b).
- (7) Error E01064: Meta-entity cannot be modified. See General Rule (7).
- (8) Warning W01037: Creation of corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity. See General Rule (6)(c) and Action (3)(c).

### 5.1.1.3 Delete Meta-Entity Command

**Function** To delete a meta-entity from the IRD Schema.

**Format**

```

delete-meta-entity-command ::=
    delete-word                4.3
    meta-entity-word           4.3
    existing-meta-entity-access-name 4.4
    [ with-meta-relationships-clause ] 6.128
    ;
    
```

**Syntax Rules** None.

**General Rules**

For the following rules:

Let *E* denote the meta-entity to be deleted.  
 Let *V* denote the effective IRD Schema view.  
 Let *P* denote the IRD Schema life-cycle-phase which contains *E*.

- (1) If with-meta-relationships-clause is specified, the meta-entity shall not participate as the second meta-entity in any meta-relationship. If with-meta-relationships-clause is not specified, it shall not participate in any meta-relationship.
- (2) The meta-entity shall not have a SYSTEM-LOCK meta-attribute equal to ON. (See the definition of the meta-attribute-type SYSTEM-LOCK in Subsection 9.3 for the rules which govern how SYSTEM-LOCK is set.)
- (3) The meta-entity shall not have an IMPLEMENTATION-LOCK meta-attribute equal to ON.
- (4) *V* shall have the attribute IRD-SCHEMA-PHASE-NAME equal to *P*.

**Security Rules** None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced. If any error was encountered, no change shall be made to the IRD Schema.
- (2) If no errors were encountered, the specified meta-entity shall be deleted. If with-meta-relationships-clause is specified, then all meta-relationships in which the specified meta-entity is the first meta-entity shall also be deleted.
- (3) Completion of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01078: Meta-entity cannot be deleted. Specified meta-entity is related to another meta-entity. See General Rule (1).

- (2) Error E01079: Cannot delete specified meta-entity. Other meta-entities depend on specified meta-entity. See General Rule (1).
- (3) Error E01073: Meta-entity is locked and cannot be deleted. See General Rules (2) and (3).

NOTE: The deactivate-IRD, activate-IRD and modify-meta-entity-life-cycle-phase commands specify rules for setting SYSTEM-LOCK and NUMBER-OF-INSTANCES meta-attributes on meta-entities for which the modify-meta-entity-life-cycle-phase command apply. For meta-entities which are created and deleted in the CONTROLLED IRD Schema life-cycle-phase, it is necessary that NUMBER-OF-INSTANCES and SYSTEM-LOCK have current values when the delete-meta-entity command is executed. This neither implies nor excludes the possibility that these meta-attributes are maintained continuously.

#### 5.1.1.4 Add Meta-Relationship Command

**Function** To create a meta-relationship.

**Format**

```

add-meta-relationship-command ::=
    add-word                4.3
    meta-relationship-word  4.3
    meta-relationship-identifier 4.7
    [ new-meta-attributes-clause ] 6.103
    ;

```

**Syntax Rules** None.

**General Rules**

For the following rules:

Let *E1* denote the first meta-entity in the meta-relationship.  
 Let *E2* denote the second meta-entity in the meta-relationship.  
 Let *P1* denote the IRD Schema life-cycle-phase in which *E1* exists.  
 Let *P2* denote the IRD Schema life-cycle-phase in which *E2* exists.  
 Let *U* denote the effective IRDS user.  
 Let *V* denote the effective IRD-schema-view.  
 Let *V'* denote an IRD-schema-view related to *U*.

- (1) Each meta-entity-access-name specified in meta-relationship-identifier shall identify an existing meta-entity.
- (2) The specified meta-relationship shall be of a valid meta-relationship-type. (See Subsection 9.2 for a table of valid meta-relationship-types.)
- (3) The specified meta-relationship shall not exist.
- (4) If the meta-relationship-type is sequenced (See Subsection 9.2), then for any given meta-entity-access-name-1 all meta-relationships of that type shall have unique sequencing-meta-attributes.
- (5) The meta-relationship shall conform to the ratio constraint for the corresponding meta-relationship-type. (See Subsection 9.2.)

- (6) If specified, the meta-attributes SINGULAR and MAXIMUM-NUMBER-OF-OCCURRENCES shall conform to the following constraints:
  - (a) MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute shall not be specified if SINGULAR equals YES or unspecified.
  - (b) If SINGULAR equals NO, MAXIMUM-NUMBER-OF-OCCURRENCES may be specified or unspecified. If unspecified, it defaults to MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT in the meta-entity EXISTING-IRDS-DEFAULTS.
- (7) If the specified meta-relationship is of type ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA, then
  - (a) The ATTRIBUTE-TYPE meta-entity specified in meta-relationship-identifier shall participate in a meta-relationship of type ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-PROCEDURE.
  - (b) If the specified ATTRIBUTE-TYPE-VALIDATION-DATA meta-entity has the VALIDATION-TYPE meta-attribute equal to VALUE, then the ATTRIBUTE-TYPE meta-entity specified in the meta-relationship-identifier shall be associated with the meta-entity VALUE-VALIDATION.
  - (c) If the specified ATTRIBUTE-TYPE-VALIDATION-DATA meta-entity has the meta-attribute VALIDATION-TYPE equal to RANGE, then the ATTRIBUTE-TYPE meta-entity specified in the meta-relationship-identifier shall be associated with the meta-entity RANGE-VALIDATION.
- (8) The specified meta-relationship shall conform to the IRD Schema life-cycle-integrity constraint (Subsection 9.8). In particular:
  - (a) Each meta-entity specified in meta-relationship-identifier shall be in either the UNCONTROLLED or CONTROLLED IRD-schema-life-cycle-phase.
  - (b) If the first meta-entity of the specified meta-relationship is in CONTROLLED phase, the second meta-entity shall be in CONTROLLED phase.
  - (c) If the meta-relationship is such that the first meta-entity is of a type for which the modify-meta-entity-life-cycle-phase-command is allowed, then the first meta-entity specified in meta-relationship-identifier shall be in UNCONTROLLED phase.
- (9) In V, the IRD-SCHEMA-PHASE-NAME attribute shall equal P1.
- (10) There shall be a V', which may be V, which has an IRD-SCHEMA-PHASE-NAME attribute equal to P2.
- (11) If the meta-relationship-type is CONTAINS, then E1 shall not be either of the following meta-entities:

DATE-TIME-ADDED  
DATE-TIME-MODIFIED

**Security Rules**     None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages are issued.
- (2) If the command passes validation, then the meta-relationship shall be created.
- (3) Completion of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01067: Meta-entity does not exist or is not visible. See General Rules (1), (9) and (10).
- (2) Error E01080: Invalid meta-relationship-type. See General Rule (2).
- (3) Error E01081: Meta-relationship already exists. See General Rule (3).
- (4) Error E01082: Conflicting sequencing meta-attributes. See General Rule (4).
- (5) Error E01083: Meta-relationship violates ratio constraint. See General Rule (5).
- (6) Error E01084: MAXIMUM-NUMBER-OF-OCCURRENCES and SINGULAR meta-attributes conflict. See General Rule (6).
- (7) Error E01085: Cannot establish meta-relationship. First meta-entity not associated with an ATTRIBUTE-TYPE-VALIDATION-PROCEDURE. See General Rule (7).
- (8) Error E01086: Cannot establish meta-relationship. Specified ATTRIBUTE-TYPE-VALIDATION-DATA not consistent with ATTRIBUTE-TYPE-VALIDATION-PROCEDURE associated with first meta-entity. See General Rule (7).
- (9) Error E01087: Creation of meta-relationship would violate IRD Schema life-cycle-phase-integrity. See General Rule (8).
- (10) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See General Rule (9).
- (11) Error E01113: Cannot establish meta-relationship. See General Rule (11).

**5.1.1.5 Modify Meta-Relationship Command**

**Function** To modify the meta-attributes of an existing meta-relationship.

**Format**

```

modify-meta-relationship-command ::=
    modify-word                4.3
    meta-relationship-identifier 4.7
    modified-meta-attributes-clause 6.104
    ;

```

**Syntax Rules** None.

## General Rules

For the following rules:

Let *E1* denote the first meta-entity in the meta-relationship.  
Let *E2* denote the second meta-entity in the meta-relationship.  
Let *P1* denote the IRD Schema life-cycle-phase in which *E1* exists.  
Let *P2* denote the IRD Schema life-cycle-phase in which *E2* exists.  
Let *U* denote the effective IRDS-user.  
Let *V* denote the effective IRD-schema-view.  
Let *V'* denote an IRD-schema-view related to *U*.

- (1) The specified meta-relationship shall exist in the IRD Schema.
- (2) If modified-meta-attributes-clause attempts to change the meta-attribute SINGULAR from NO to YES, then the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute shall be deleted.
- (3) If the first meta-entity in the specified meta-relationship is of a type for which the modify-meta-entity-life-cycle-phase command applies, then it shall be in the UNCONTROLLED IRD Schema life-cycle-phase.
- (4) The first meta-entity in the specified meta-relationship shall not be in the ARCHIVED IRD Schema life-cycle-phase.
- (5) In *V*, the IRD-SCHEMA-PHASE-NAME attribute shall equal *P1*.
- (6) There shall be an IRD-schema-view *V'*, which may be *V*, which has an IRD-SCHEMA-PHASE-NAME attribute equal to *P2*.
- (7) If the meta-relationship-class-type is CONTAINS, then *E1* shall not be either of the following meta-entities:

DATE-TIME-ADDED  
DATE-TIME-MODIFIED

**Security Rules**     None.

## Actions

- (1) The command shall be validated. All appropriate error and warning messages shall be issued.
- (2) If no errors were encountered, then the meta-attributes shall be modified as specified.
- (3) Completion of command processing shall be confirmed.

## Error and Warning Conditions

- (1) Error E01088: Meta-relationship does not exist or is not completely visible. See General Rules (1), (5), and (6).
- (2) Error E01084: MAXIMUM-NUMBER-OF-OCCURRENCES and SINGULAR meta-attributes conflict. See General Rule (2).



- (3) Error E01089: Meta-relationship maintenance would impact a non-modifiable meta-entity. See General Rules (3), (4), and (7).
- (4) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See General Rule (5).

### 5.1.1.6 Delete Meta-Relationship Command

**Function** To delete a meta-relationship.

**Format**

```
delete-meta-relationship-command ::=
    delete-word                4.3
    meta-relationship-word      4.3
    meta-relationship-identifier 4.7
    ;
```

**Syntax Rules** None.

**General Rules**

For the following rules:

Let *E1* denote the first meta-entity in the meta-relationship.  
 Let *E2* denote the second meta-entity in the meta-relationship.  
 Let *P1* denote the IRD Schema life-cycle-phase in which *E1* exists.  
 Let *P2* denote the IRD Schema life-cycle-phase in which *E2* exists.  
 Let *U* denote the effective IRDS-user.  
 Let *V* denote the effective IRD-schema-view.  
 Let *V'* denote an IRD-schema-view related to *U*.

- (1) The specified meta-relationship shall exist in the IRD Schema.
- (2) If both meta-entities in the specified meta-relationship are in the CONTROLLED IRD Schema life-cycle-phase, the result of this command shall not invalidate the contents of the IRD.
- (3) This command shall not be allowed if the SYSTEM-LOCK or IMPLEMENTATION-LOCK meta-attribute of the specified meta-relationship equals ON. (See the description of these meta-attribute-types for the rules for the setting of these meta-attributes.)
- (4) In *V*, the IRD-SCHEMA-PHASE-NAME attribute shall equal *P1*.
- (5) There shall be an IRD-schema-view *V'*, which may be *V*, which has an IRD-SCHEMA-PHASE-NAME attribute equal to *P2*.
- (6) If the meta-relationship-class-type is CONTAINS, then *E1* shall not be either of the following meta-entities:

DATE-TIME-ADDED  
 DATE-TIME-MODIFIED

**Security Rules** None.

**Actions**

- (1) The command shall be validated. All appropriate error messages and diagnostics shall be issued.
- (2) If no errors were encountered, then the meta-relationship shall be deleted.
- (3) Completion of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01088: Meta-relationship does not exist or is not completely visible. See General Rules (1), (4), and (5).
- (2) Error E01090: Maintenance would result in an inconsistency between the IRD and the IRD Schema. See General Rule (2).
- (3) Error E01091: Meta-relationship is locked and cannot be deleted. See General Rule (3).
- (4) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See General Rule (4).
- (5) Error E01089: Meta-relationship maintenance would impact a non-modifiable meta-entity. See General Rule (6).

**5.1.1.7 Modify Meta-Entity Access-Name Command**

**Function** To change the assigned-access-name of a meta-entity.

**Format**

```

modify-meta-entity-access-name-command ::=
    modify-word           4.3
    meta-entity-word     4.3
    access-name-word     4.3
    from-word            4.3
    existing-meta-entity-assigned-access-name
    to-word              4.3
    new-meta-entity-assigned-access-name
    ;

existing-meta-entity-assigned-access-name ::=
    assigned-access-name 4.4

new-meta-entity-assigned-access-name ::=
    assigned-access-name 4.4
    
```

**Syntax Rules** None.

**General Rules**

- (1) Existing-meta-entity-assigned-access-name shall be the assigned-access-name of at least one existing meta-entity in the IRD Schema.

- (2) New-meta-entity-assigned-access-name shall not be the same as:
  - (a) The assigned-access-name of an existing meta-entity;
  - (b) The assigned-descriptive-name of an existing meta-entity other than those meta-entities identified by existing-meta-entity-assigned-access-name;
- (3) Meta-entity-assigned-access-name shall not identify any meta-entity with a reserved-name. Reserved meta-entity-assigned-access-names are defined in the IRD Schema in IRDS-RESERVED-NAMES meta-entities. Each reserved assigned access-name equals an occurrence of a RESERVED-META-ENTITY-NAME meta-attribute in one IRDS-RESERVED-NAMES meta-entity.
- (4) If this command changes the name of any meta-entity in the CONTROLLED IRD Schema life-cycle-phase, it shall be executed only when the IRD is deactivated. (See Subsections 5.1.1.11 and 5.1.1.13).
- (5) If this command changes the name of any meta-entity in the CONTROLLED IRD Schema life-cycle-phase, it shall be issued by the same IRDS-user who issued the deactivate-IRD command.

**Security Rules**     None.

#### **Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) If no errors were encountered, then
  - (a) For each meta-entity which has an assigned-access-name equal to existing-meta-entity-assigned-access-name, the existing-meta-entity-assigned-access-name shall be changed to new-meta-entity-assigned-access-name.
  - (b) For each meta-entity for which the assigned-access-name is changed, the following meta-attributes shall be modified as specified below:

NUMBER-OF-TIMES-MODIFIED = prior-value + 1  
 LAST-MODIFIED-BY = name of effective IRDS-user

The meta-attribute-group DATE-TIME-MODIFIED shall be changed as follows:

SYSTEM-DATE = date of transaction  
 SYSTEM-TIME = time of transaction

- (3) Completion of the command shall be confirmed.

#### **Error and Warning Conditions**

- (1) Error E01092: No meta-entity exists with specified name. See General Rule (1).
- (2) Error E01093: New name conflicts with an assigned-access-name of an existing meta-entity. See General Rule (2)(a).

- (3) Error E01094: New name conflicts with an assigned-descriptive-name of existing meta-entity. See General Rule (2)(b).
- (4) Error E01097: Assigned-access-name is reserved and cannot be changed. See General Rule (3).
- (5) Error E01098: IRD must be deactivated. See General Rule (4).
- (6) Error E01099: Another user deactivated the IRD. See General Rule (5).
- (7) Error E01114: Assigned-access-name cannot be changed. See General Rule (6).

**5.1.1.8 Modify Meta-Entity Descriptive-Name Command**

**Function** To change the assigned-descriptive-name of a meta-entity.

**Format**

```

modify-meta-entity-descriptive-name-command ::=
    modify-word                4.3
    meta-entity-word           4.3
    descriptive-name-word      4.3
    existing-meta-entity-specification
    to-word                    4.3
    new-meta-entity-assigned-descriptive-name
    ;

existing-meta-entity-specification ::=
    using-meta-entity-assigned-descriptive-
        name-specification
    | using-meta-entity-assigned-access-name-
        specification

using-meta-entity-assigned-descriptive-
    name-specification ::=
    from-word                  4.3
    existing-meta-entity-assigned-
        descriptive-name

using-meta-entity-assigned-access-name-
    specification ::=
    for-word                   4.3
    existing-meta-entity-assigned-access-
        name

existing-meta-entity-assigned-descriptive-
    name ::=
    assigned-descriptive-name  4.4

existing-meta-entity-assigned-access-name ::=
    assigned-access-name      4.4

new-meta-entity-assigned-descriptive-name ::=
    assigned-descriptive-name  4.4
    
```

**Syntax Rules**    None.

**General Rules**

- (1) If using-meta-entity-assigned-descriptive-name-specification is used, then existing-meta-entity-assigned-descriptive-name shall be the assigned-descriptive-name of at least one existing meta-entity in the IRD Schema.  
  
If using-meta-entity-assigned-access-name-specification is used, then existing-meta-entity-assigned-access-name shall be the assigned-access-name of at least one existing meta-entity in the IRD Schema, and the descriptive-name of each such meta-entity shall be null.
- (2) New-meta-entity-assigned-descriptive-name shall not be the same as:
  - (a) The assigned-access-name of an existing meta-entity other than those meta-entities identified by existing-meta-entity-assigned-descriptive-name;
  - (b) The assigned-descriptive-name of an existing meta-entity;
- (3) If this command changes the name of any meta-entity in the CONTROLLED IRD Schema life-cycle-phase, it shall be executed only when the IRD is deactivated. (See Subsections 5.1.1.11 and 5.1.1.13.)
- (4) If this command changes the name of any meta-entity in the CONTROLLED IRD Schema life-cycle-phase, it shall be issued by the same IRDS-user who issued the deactivate-IRD command.

**Security Rules**    None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) If no errors were encountered, then
  - (a) Each meta-entity with an assigned-descriptive-name of existing-meta-entity-assigned-descriptive-name shall be changed to have an assigned descriptive-name of new-meta-entity-assigned-descriptive-name.
  - (b) Each meta-entity with an assigned-access-name of existing-meta-entity-assigned-access-name shall be changed to have an assigned-descriptive-name of new-meta-entity-assigned-descriptive-name.
  - (c) For each meta-entity for which the assigned-descriptive-name shall be changed, the following meta-attributes shall be modified as specified below:

NUMBER-OF-TIMES-MODIFIED = prior-value + 1  
DATE-TIME-MODIFIED = name of effective IRDS-user

The meta-attribute-group DATE-TIME-MODIFIED shall be changed as follows:

SYSTEM-DATE = date of transaction

SYSTEM-TIME = time of transaction

- (3) Completion of the command shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01092: No meta-entity exists with specified name. See General Rule (1).
- (2) Error E01094: New name conflicts with an assigned-descriptive-name of an existing meta-entity. See General Rule (2)(a).
- (3) Error E01093: New name conflicts with an assigned-access-name of an existing meta-entity. See General Rule (2)(b).
- (5) Error E01098: IRD must be deactivated. See General Rule (3).
- (6) Error E01099: Another user deactivated the IRD. See General Rule (4).

**5.1.1.9 Modify Meta-Entity Life-Cycle-Phase Command**

**Function** To move a meta-entity from one IRD Schema life-cycle-phase to another.

**Format**

```

modify-meta-entity-life-cycle-phase-command ::=
    modify-word           4.3
    meta-entity-word      4.3
    life-cycle-phase-word 4.3
    [ for-word ]         4.3
    meta-entity-name-list
    from-word            4.3
    current-IRD-schema-life-cycle-phase
    to-word              4.3
    new-IRD-schema-life-cycle-phase
    [ implementation-defined-options ]
    ;

meta-entity-name-list ::=
    meta-entity-access-name 4.4
    [ , meta-entity-name-list ]

current-IRD-schema-life-cycle-phase ::=
    IRD-schema-life-cycle-phase 4.7

new-IRD-schema-life-cycle-phase ::=
    IRD-schema-life-cycle-phase 4.7
    
```

**Syntax Rules** None.

**General Rules**

For the following rules:

Let *E* denote a meta-entity.

Let  $X$  denote a meta-relationship-class-type  
 Let  $E'$  denote a meta-entity for which the meta-relationship  $E' X E$  exists.  
 Let  $E''$  denote a meta-entity for which the meta-relationship  $E X E''$  exists.  
 Let  $P1$  be current-IRD-schema-life-cycle-phase.  
 Let  $P2$  be new-IRD-schema-life-cycle-phase.  
 Let  $U$  be the effective IRDS-user.  
 Let  $V$  be the effective IRD-schema-view.  
 Let  $V'$  be an IRD-schema-view related to  $U$ .

- (1) This command shall be executed only when the IRD is deactivated. See Subsections 5.1.1.11 and 5.1.1.13.
- (2) The IRDS-user who issues this command shall be the same IRDS-user who issued the deactivate-IRD command.
- (3) The allowed combinations of current-IRD-schema-life-cycle-phase and new-IRD-schema-life-cycle-phase shall be as follows:

current IRD Schema <u>life-cycle-phase</u>	new IRD Schema <u>life-cycle-phase</u>
UNCONTROLLED CONTROLLED	CONTROLLED ARCHIVED
current IRD Schema <u>life-cycle-phase</u>	new IRD Schema <u>life-cycle-phase</u>
ARCHIVED CONTROLLED	CONTROLLED UNCONTROLLED

- (4) The IRD-SCHEMA-PHASE-NAME attribute for  $V$  shall be  $P2$ .
- (5) There shall be an IRD-schema-view  $V'$  related to  $U$  with an IRD-SCHEMA-PHASE-NAME attribute equal to  $P1$ .
- (6) Each meta-entity specified in meta-entity-name-list shall exist in the IRD-schema.
- (7) Each meta-entity specified in meta-entity-name-list shall exist in current-IRD-schema-life-cycle-phase.
- (8) Each meta-entity specified in meta-entity-name-list shall be of one of the following types:

RELATIONSHIP-TYPE  
 RELATIONSHIP-CLASS-TYPE  
 ENTITY-TYPE  
 VARIATION-NAMES-DATA  
 ATTRIBUTE-GROUP-TYPE  
 ATTRIBUTE-TYPE  
 ATTRIBUTE-TYPE-VALIDATION-DATA

- (9) The command shall not violate IRD Schema life-cycle-phase integrity (Subsection 9.8). For this command, it may be restated as follows:

- (a) If *E* is moved from the UNCONTROLLED to CONTROLLED IRD Schema life-cycle-phase, then:
  - (i) *E'* shall be in the UNCONTROLLED IRD Schema life-cycle-phase; and
  - (ii) *E''* shall be in the CONTROLLED IRD Schema life-cycle-phase.
- (b) If *E* is moved from the CONTROLLED to ARCHIVED IRD Schema life-cycle-phase, then:
  - (i) *E'* shall be in the ARCHIVED IRD Schema life-cycle-phase; and
  - (ii) *E''* shall be in the CONTROLLED IRD Schema life-cycle-phase.
- (c) If *E* is moved from the ARCHIVED to CONTROLLED IRD Schema life-cycle-phase, then:
  - (i) *E'* shall be in the ARCHIVED IRD Schema life-cycle-phase; and
  - (ii) *E''* shall be in the CONTROLLED IRD Schema life-cycle-phase.
- (d) If *E* is moved from the CONTROLLED to UNCONTROLLED IRD Schema life-cycle-phase, then:
  - (i) *E'* shall be in the UNCONTROLLED IRD Schema life-cycle-phase,
  - (ii) *E''* shall be in the CONTROLLED IRD Schema life-cycle-phase.

**Security Rules**     None.

**Actions**

- (1) The command syntax shall be validated. If the command syntax is valid, conformance with General Rules (1) through (5) shall be checked. If the command passes this validation, then General Rules (6) through (8) shall be checked for each meta-entity specified in meta-entity-name-list.
- (2) If any errors were encountered, all appropriate error messages shall be produced.
- (3) If no errors were encountered, each meta-entity specified in meta-entity-list shall be processed as follows:
  - (a) Conformance with General Rule (9) shall be checked.
  - (b) If the meta-entity passes General Rule (9), it shall be moved from current-IRD-schema-life-cycle-phase to new-IRD-schema-life-cycle-phase. The audit meta-attributes and meta-attribute-groups shall be modified as described in the modify-meta-entity command.

The SYSTEM-LOCK meta-attribute shall be maintained according to the following rules:

- (i) Whenever a meta-entity is moved into the CONTROLLED IRD Schema life-cycle-phase, SYSTEM-LOCK shall be set equal to ON.



- (ii) Whenever a meta-entity is moved out of the CONTROLLED IRD Schema life-cycle-phase, SYSTEM-LOCK shall be set equal to OFF, provided that the NUMBER-OF-INSTANCES meta-attribute shall not be greater than zero. (See the deactivate-IRD and activate-IRD commands for rules regarding the NUMBER-OF-INSTANCES meta-attributes.)

Processing continues with the next meta-entity.

- (c) If the meta-entity-fails General Rule (7), all appropriate error messages shall be produced. Processing continues with the next meta-entity.
- (4) If *E* is moved from the UNCONTROLLED to CONTROLLED IRD Schema life-cycle-phase, then if:
  - (a) *E* is of type ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE, and
  - (b) *E* has the meta-attribute of type COMMON-TO-ENTITY-TYPES equal to YES, meta-relationships between *E* and all meta-entities of type ENTITY-TYPE in the CONTROLLED IRD Schema life-cycle-phase shall be established.
- (5) Completion of command processing shall be confirmed.

#### **Error and Warning Conditions**

- (1) Error E01098: IRD must be deactivated. See General Rule (1).
- (2) Error E01099: Another user deactivated the IRD. See General Rule (2).
- (3) Error E01100: Invalid phase transition. See General Rule (3).
- (4) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See General Rule (4).
- (5) Error E01067: Meta-entity does not exist or is not visible. See General Rules (5) and (6).
- (6) Error E01101: Meta-entity not in current IRD Schema life-cycle-phase. See General Rule (7).
- (7) Error E01102: Meta-entity is of type which is not valid for command. See General Rule (8).
- (8) Error E01103: Meta-entity cannot be moved. It would result in meta-relationships which violate IRD Schema life-cycle-phase integrity. See General Rule (9).

#### **NOTES:**

- (1) Movement of meta-entities is very sensitive with respect to order. An implementation option may sort the meta-entities by type after completion of Action (2) so that the command may have greater assurance of successful completion. The following sort sequence is suggested to minimize possible errors:

If the movement is to the CONTROLLED IRD-schema-life-cycle-phase:

ATTRIBUTE-TYPE-VALIDATION-DATA meta-entities should precede ATTRIBUTE-TYPE meta-entities.

ATTRIBUTE-TYPE-VALIDATION-DATA and ATTRIBUTE-TYPE meta-entities should precede ATTRIBUTE-GROUP-TYPE meta-entities.

ATTRIBUTE-TYPE-VALIDATION-DATA, ATTRIBUTE-TYPE, ATTRIBUTE-GROUP-TYPE, and VARIATION-NAMES-DATA meta-entities should precede ENTITY-TYPE meta-entities.

ATTRIBUTE-TYPE-VALIDATION-DATA, ATTRIBUTE-TYPE, ATTRIBUTE-GROUP-TYPE, VARIATION-NAMES-DATA, ENTITY-TYPE, and RELATIONSHIP-CLASS-TYPE meta-entities should precede RELATIONSHIP-TYPE meta-entities.

For the movements from CONTROLLED, the order should be reversed.

Note that if an implementation does not sort meta-entities as specified, it becomes the user's responsibility to specify the proper order of movement.

- (2) An implementation-option may reject the entire command if any error is encountered on any meta-entity in Action (3).

### 5.1.1.10 Copy Meta-Entity Command

**Function** To create a meta-entity with the same user-specifiable meta-attributes as an existing meta-entity.

#### Format

```

copy-meta-entity-command ::=
    copy-word                4.3
    meta-entity-word         4.3
    existing-meta-entity-access-name 4.4
    [ with-meta-relationships-clause ] 6.128
    to-word                  4.3
    new-meta-entity-specification
    [ meta-entity-descriptive-name-clause ] 6.126
    ;

new-meta-entity-specification ::=
    new-meta-entity-access-name 4.4
    | new-meta-entity-version-clause 6.127
    
```

**Syntax Rules** None.

#### General Rules

For the following Rules:

- Let *E* be the meta-entity identified by existing-meta-entity-access-name.
- Let *T* be the type of *E*.
- Let *E'* be the meta-entity identified by new-meta-entity-specification.
- Let *P* be the IRD-schema-life-cycle-phase in which *E* exists.
- Let *V* be the effective IRD-schema-view.
- Let *V'* be an IRD-schema-view which is related to *U*.
- Let *U* be the effective IRDS-user.

- (1) This command shall be valid only for meta-entities of the following types:

ATTRIBUTE-GROUP-TYPE  
 ATTRIBUTE-TYPE  
 ATTRIBUTE-TYPE-VALIDATION-DATA  
 ENTITY-TYPE  
 RELATIONSHIP-TYPE  
 RELATIONSHIP-CLASS-TYPE  
 VARIATION-NAMES-DATA

- (2) If with-meta-relationships-clause is specified, then for each meta-relationship in which *E* is the first meta-entity, a corresponding meta-relationship with

- (a) The new meta-entity as the first meta-entity,  
 (b) The same second meta-entity, and  
 (c) The same meta-attributes,

shall be created unless the corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity.

- (3) The assigned-access-name in new-meta-entity-access-name shall not be same as:

- (a) The assigned-access-name of an existing meta-entity.  
 (b) The assigned-descriptive-name of an existing meta-entity.

- (4) Meta-entity-descriptive-name-clause shall not be allowed if new-meta-entity-version-clause is specified and the meta-entity specified by existing-meta-entity-access-name has a descriptive-name.

- (5) *V* shall have the IRD-SCHEMA-PHASE-NAME attribute equal to UNCONTROLLED.

- (6) There shall be an IRD-schema-view *V'*, which may be *V*, which has an IRD-SCHEMA-PHASE-NAME attribute equal to *P*.

- (7) If *E* is one of the following meta-entities:

DATE-TIME-ADDED  
 DATE-TIME-MODIFIED  
 ADDED-BY  
 MODIFIED-BY  
 NUMBER-OF-TIMES-MODIFIED  
 SYSTEM-DATE  
 SYSTEM-TIME

then new-meta-entity-version-clause shall not be used.

**Security Rules**    None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) If no errors are encountered, then:
  - (a) A new meta-entity with the specified access-name shall be created in the UNCONTROLLED IRD Schema life-cycle-phase.
    - (i) If meta-entity-descriptive-name-clause is specified, the new meta-entity shall be given the assigned-descriptive-name specified.
    - (ii) If meta-entity-descriptive-name-clause is not specified and new-meta-entity-version-clause is specified, then the new meta-entity shall be given a descriptive-name with the same assigned-descriptive-name as the meta-entity being copied.
    - (iii) If meta-entity-descriptive-name-clause and new-meta-entity-version-clause are not specified, then no descriptive-name shall be assigned to the new meta-entity.
  - (b) All user-specifiable meta-attributes of the meta-entity identified by existing-meta-entity-access-name shall be assigned to the meta-entity identified by new-meta-entity-specification.
  - (c) The system-maintained meta-attributes as well as the meta-attribute-groups for the meta-entity identified by new-meta-entity-specification shall be assigned as in add-meta-entity-command.
  - (d) If meta-entity-descriptive-name-clause and new-meta-entity-version-clause are specified, then for each meta-entity which existed prior to execution of this command and has the same assigned-access-name as specified in existing-meta-entity-access-name:
    - (i) The meta-entity shall be given the appropriate descriptive-name.
    - (ii) The system-maintained meta-attributes shall be changed as in the modify-meta-entity-descriptive-name-command.
  - (e) Let  $E$  be the meta-entity identified by existing-meta-entity-access-name, and  $N$  be the new meta-entity. Let  $E'$  denote any other meta-entity and  $X$  a meta-relationship-class-type. Then for every meta-relationship  $E X E'$ ; a corresponding meta-relationship  $N X E'$  is created, provided it does not violate IRD Schema life-cycle-phase integrity. For each such meta-relationship which cannot be created, a warning shall be issued.
- (3) If  $T$  is ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE, and  $E$  has the meta-attribute of type COMMON-TO-ENTITY-TYPES equal to YES, then  $E'$  shall have this meta-attribute equal to NO.
- (4) Completion of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01102: Meta-entity is of type which is not valid for command. See General Rule (1).
- (2) Warning W01037: Creation of corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity. See General Rule (2) and Action (2)(e).
- (3) Error E01093: New name conflicts with an assigned-access-name of an existing meta-entity. See General Rule (3)(a).
- (4) Error E01094: New name conflicts with an assigned-descriptive name of existing meta-entity. See General Rule (3)(b).
- (5) Error E01075: Meta-entity already has a descriptive-name. See General Rule (4).
- (6) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase. See General Rule (5).
- (7) Error E01067: Meta-entity does not exist or is not visible. See General Rule (6).
- (8) Error E01096: New-meta-entity-version-clause not allowed for specified meta-entity. See General Rule (7).

**5.1.1.11 Deactivate IRD Command**

**Function** To stop all IRD activity and restrict access to the IRD Schema to a single user.

**Format**

```

deactivate-IRD-command ::=
    deactivate-word           4.3
    IRD-word                 4.3
    [ implementation-defined-options ]
    ;

```

**Syntax Rules** None.

**General Rules**

- (1) No other deactivate-IRD-command shall be in effect.

**Security Rules** None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) If no errors were encountered, then the following commands shall be disabled for all users of the IRD being deactivated:
  - (a) All IRD commands.
  - (b) All utility commands except exit-IRDS-system-command.

- (c) Export-IRD-command.
- (d) Check-IRD-schema-compatibility-command
- (e) Import-IRD-command.
- (f) IRD-schema-output-command. (An implementation may, optionally, allow use of this command to the IRDS user who issued the deactivate-IRD command.)

No new commands of the above list shall be allowed to enter execution.

All other commands shall be restricted to the IRDS-user who issued the deactivate-IRD command.

Once all the commands have been either disabled or restricted as specified above, and there are no active transactions for any user other than the IRDS-user who issued the deactivate-IRD-command, the IRD is said to have attained a deactivated state. The IRD shall remain in a deactivated state until the IRDS-user who issued the deactivate-IRD command executes an activate-IRD command.

The implementation shall ensure that the IRD Schema is recoverable from its state as of the termination of all IRD activity.

- (3) Completion of command processing shall be confirmed.

### **Error and Warning Conditions**

- (1) Error E01104: IRD is already deactivated.

#### **NOTES:**

- (1) The implementation may add options such as:
  - (a) Message text to be broadcast to all active users.
  - (b) An effective shutdown date and time.
  - (c) Forced termination of active transactions.
- (2) For each meta-entity of a type for which the modify-meta-entity-life-cycle-phase command is valid, it shall only be necessary that the NUMBER-OF-INSTANCES meta-attribute be correct when the IRD is deactivated.
- (3) No new user shall become active once the command has been issued.

### **5.1.1.12 Restore IRD Schema Command**

**Function** To restore the IRD Schema to its state as of the last time a deactivate-IRD command was issued.

#### **Format**

```
restore-IRD-schema-command ::=
    restore-word                4.3
    IRD-schema-word            4.3
    ;
```

**Syntax Rules** None.

**General Rules**

- (1) The IRD shall be deactivated.
- (2) The IRDS-user who issues this command shall be the same IRDS-user who issued the deactivate-IRD command.

**Security Rules** None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) If no errors were encountered, the IRD Schema shall be restored to its state as of the execution of the last deactivate-IRD-command.
- (3) At completion of the command processing the IRD is still deactivated.
- (4) Completion of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01098: IRD must be deactivated. See General Rule (1).
- (2) Error E01099: Another user deactivated the IRD. See General Rule (2).

**5.1.1.13 Activate IRD Command**

**Function** To enable IRD commands.

**Format**

```

activate-IRD-command ::=
    activate-word           4.3
    IRD-word               4.3
    [ implementation-defined-options ]
    ;
    
```

**Syntax Rules** None.

**General Rules**

- (1) The IRD shall be in a deactivated state.
- (2) The IRDS-user who issues this command shall be the same IRDS-user who issued the deactivate IRD command.
- (3) The meta-entities in the CONTROLLED IRD Schema life-cycle-phase shall conform to the following rules:
  - (a) Each meta-entity shall have a unique assigned-access-name.

- (b) Each RELATIONSHIP-TYPE meta-entity shall have exactly two meta-relationships of the type RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE.
- (c) For each RELATIONSHIP-CLASS-TYPE meta-entity *C*, there shall be a single meta-entity *R* in the CONTROLLED IRD Schema life-cycle-phase such that the meta-relationship *R* MEMBER-OF *C* exists.
- (d) For each ATTRIBUTE-TYPE meta-entity *A*, there shall be a meta-entity *E* of type ATTRIBUTE-GROUP-TYPE, ENTITY-TYPE, or RELATIONSHIP-TYPE also in the CONTROLLED IRD Schema life-cycle-phase, such that there exists a meta-relationship *E* CONTAINS *A*.
- (e) For each ATTRIBUTE-GROUP-TYPE meta-entity *G*, there shall be a meta-entity *E* of type ENTITY-TYPE or RELATIONSHIP-TYPE also in the CONTROLLED IRD Schema life-cycle-phase, such that there exists a meta-relationship *E* CONTAINS *G*.
- (f) If *A* is an ATTRIBUTE-TYPE meta-entity for which the meta-attribute of type COMMON-TO-ENTITY-TYPES is YES, and *E* is a meta-entity of type ENTITY-TYPE in the CONTROLLED IRD Schema life-cycle-phase, there shall exist a meta-relationship *E* CONTAINS *A*.
- (g) If *G* is an ATTRIBUTE-GROUP-TYPE meta-entity for which the meta-attribute of type COMMON-TO-ENTITY-TYPES is YES, and *E* is a meta-entity of type ENTITY-TYPE in the CONTROLLED IRD Schema life-cycle-phase, there shall exist a meta-relationship *E* CONTAINS *G*.
- (h) For each ATTRIBUTE-TYPE-VALIDATION-DATA meta-entity *D* there shall be a meta-entity *A* of type ATTRIBUTE-TYPE in the CONTROLLED phase such that the meta-relationship *A* USES *D* exists.
- (i) For each ATTRIBUTE-GROUP-TYPE meta-entity *G*:
  - (i) There shall be at least one ATTRIBUTE-TYPE meta-entity *A* for which the meta-relationship *G* CONTAINS *A* exists.
  - (ii) The meta-attribute SIGNIFICANT-ATTRIBUTES for *G* shall be less than or equal to the number of meta-relationships of type ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE in which *G* participates.
- (j) No two meta-entities in the CONTROLLED IRD Schema life-cycle-phase shall have the same substitute-name.
- (k) No two meta-entities in the CONTROLLED IRD Schema life-cycle-phase shall have the same inverse-name.
- (l) The inverse-name of a meta-entity shall not be the same as the substitute-name of any meta-entity in the CONTROLLED IRD Schema life-cycle-phase.
- (m) The inverse-name of a meta-entity shall not be the same as the assigned-access-name or assigned-descriptive-name of any meta-entity in the CONTROLLED IRD Schema life-cycle-phase.
- (n) The substitute-name of a meta-entity shall not be the same as the assigned-access-



name or assigned-descriptive-name of any meta-entity in the CONTROLLED IRD Schema life-cycle-phase.

- (4) The contents of the IRD Schema in the CONTROLLED IRD Schema life-cycle-phase shall be compatible with the contents of the IRD. That is:
- (a) Each entity, relationship, attribute-group, and attribute shall be an instance of a type defined by a corresponding meta-entity in the CONTROLLED IRD Schema life-cycle-phase.
  - (b) Let  $A$  be an ATTRIBUTE-TYPE meta-entity in the CONTROLLED IRD Schema life-cycle-phase. Let  $a$  be an attribute of type  $A$ . If there is a non-empty set  $\{AV_1 \dots AV_n\}$  of ATTRIBUTE-TYPE-VALIDATION-DATA meta-entities which are associated with  $A$ , let  $V_1 \dots V_n$  be the set of values defined by  $AV_1$  through  $AV_n$  respectively. Let  $V$  denote the union of all  $V_1$  through  $V_n$ , and let  $v$  be a value in  $V$ . Then  $a = v$  for some  $v$  in  $V$ .
  - (c) Let  $g$  be an attribute-group with component attributes  $a_i$ ,  $1 \leq i \leq n$ . Then:
    - (i)  $g$  shall be an instance of an ATTRIBUTE-GROUP-TYPE meta-entity  $G$  in the CONTROLLED IRD Schema life-cycle-phase.
    - (ii) Each  $a_i$ ,  $1 \leq i \leq n$ , shall be an instance of a corresponding ATTRIBUTE-TYPE meta-entity  $A_i$  in the CONTROLLED IRD-schema-life-cycle-phase.
    - (iii) There shall be a meta-relationship  $G$  CONTAINS  $A_i$  for  $G$  and each  $A_i$  as above.
  - (d) If  $e$  is an entity of type  $E$ , and  $a$  is an attribute of type  $A$  such that  $a$  is an attribute of  $e$ , then the meta-relationship  $E$  CONTAINS  $A$  shall exist in the IRD Schema.
  - (e) If  $e$  is an entity of type  $E$ , and  $g$  is an attribute-group of type  $G$  such that  $g$  is declared in  $e$ , then the meta-relationship  $E$  CONTAINS  $G$  shall exist in the IRD Schema.
  - (f) If  $r$  is a relationship of type  $R$ , and  $a$  is an attribute of type  $A$  such that  $a$  is an attribute of  $R$ , then the meta-relationship  $R$  CONTAINS  $A$  shall exist in the IRD Schema.
  - (g) If  $r$  is a relationship of type  $R$ , and  $g$  is an attribute-group of type  $G$  such that  $g$  is declared in  $r$ , then the meta-relationship  $R$  CONTAINS  $G$  shall exist in the IRD Schema.
  - (h) If  $x$  is an entity or relationship of type  $X$ , and  $y_1 \dots y_n$  are attributes or attribute-groups of type  $Y$ , and  $Y$  repeats in  $X$ , then there shall not be more instances of  $Y$  in  $X$  than allowed by the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute of the meta-relationship  $X$  CONTAINS  $Y$ .
  - (i) If  $G$  is an ATTRIBUTE-GROUP-TYPE in the CONTROLLED IRD Schema life-cycle-phase which repeats within an ENTITY-TYPE  $E$  or a RELATIONSHIP-TYPE  $R$ , then there shall not exist two instances of  $G$  within any given entity of type  $E$  or relationship of type  $R$  with the same key attributes.

- (j) If  $A$  is a key ATTRIBUTE-TYPE for the ATTRIBUTE-GROUP-TYPE  $G$ , and both are in the CONTROLLED IRD Schema life-cycle-phase, then within each instance of  $G$  there shall be a corresponding instance of  $A$ .
- (k) Let  $R$  be a relationship-type in the CONTROLLED IRD Schema life-cycle-phase, and let  $A$  be an ATTRIBUTE-TYPE for which the meta-relationship  $R$  CONTAINS  $A$  exists with the meta-attribute SEQUENCE-PARAMETER = YES. Let  $e R e1$  and  $e R e2$  be instances of  $R$  with the same first entity, with  $a1$  and  $a2$  the respective instances of  $A$ . Then  $a1$  does not equal  $a2$ .
- (l) Let  $R$  and  $A$  be as above, except that on the meta-relationship  $R$  CONTAINS  $A$ , the meta-attribute SEQUENCE-PARAMETER = NO. Then there shall not exist two instances of  $R$  with the same ordered pair of entities.
- (m) If  $E$  is an ENTITY-TYPE meta-entity in the CONTROLLED IRD Schema life-cycle-phase, and  $A(V:N)$  is the access-name of an entity of type  $E$ , then the variation-name  $V$  shall be equal to a VARIATION meta-attribute in a VARIATION-NAMES-DATA meta-entity which is associated with  $E$ .
- (n) The assigned-access-name and assigned-descriptive-name of each entity in the IRD shall conform to the length and picture constraints defined in the corresponding ENTITY-TYPE meta-entity in the CONTROLLED IRD Schema life-cycle-phase. If such a constraint is not defined in the corresponding entity-type, the name shall conform to the corresponding constraint defined in EXISTING-IRDS-DEFAULTS.
- (o) Each attribute shall conform to the format, length, and picture constraints defined in the corresponding ATTRIBUTE-TYPE meta-entity in the CONTROLLED IRD Schema life-cycle-phase. If such a constraint is not defined in the corresponding entity-type, the attribute shall conform to the corresponding constraint defined in EXISTING-IRDS-DEFAULTS.

**Security Rules**     None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) The meta-attributes of type SYSTEM-LOCK and NUMBER-OF-INSTANCES are modified according to the following rules:
  - (a) Let  $E(n1)$  and  $E(n2)$  denote two meta-entities with the same assigned-access-name. Assume that  $E(n1)$  is in the CONTROLLED IRD Schema life-cycle-phase and  $E(n2)$  is in another IRD Schema life-cycle-phase. Also assume that  $E(n1)$  has the meta-attribute NUMBER-OF-INSTANCES equal to zero, and that  $E(n2)$  has a NUMBER-OF-INSTANCES meta-attribute greater than zero. Then:
    - (i) The NUMBER-OF-INSTANCES meta-attribute of  $E(n1)$  shall be set equal to the NUMBER-OF-INSTANCES meta-attribute of  $E(n2)$ ; and
    - (ii) The NUMBER-OF-INSTANCES meta-attribute of  $E(n2)$  shall be set to zero.
  - (b) SYSTEM-LOCK meta-attributes shall be set as needed in meta-entities and meta-

relationships. See Subsection 9.3 for the rules which determine how SYSTEM-LOCK is set.

- (3) If no errors are encountered, the following commands shall be enabled for the IRD which is activated by this command:
  - (a) All IRD commands.
  - (b) All utility commands.
  - (c) Export-IRD-command.
  - (d) Check-IRD-schema-compatibility-command.
  - (e) Import-IRD-command.
  - (f) IRD Schema output.
- (4) Completion of command processing shall be confirmed. The IRD is no longer in a disabled state. Normal IRD activity can resume.

### **Error and Warning Conditions**

- (1) Error E01900: Two meta-entities in CONTROLLED phase have the same assigned-access-name. See General Rule (3)(a).
- (2) Error E01901: Incomplete relationship-type. See General Rule (3)(b).
- (3) Error E01902: Unused meta-entity in CONTROLLED phase. See General Rules (3)(c), (3)(d), (3)(e), and (3)(f).
- (4) Error E01903: Attribute-group-type has an incomplete key. See General Rule (3)(g).
- (5) Error E01904: Two CONTROLLED meta-entities have the same substitute-name. See General Rule (3)(h).
- (6) Error E01905: Two CONTROLLED meta-entities have the same inverse-name. See General Rule (3)(i).
- (7) Error E01906: Inverse-name of CONTROLLED meta-entity is same as the substitute-name of a CONTROLLED meta-entity. See General Rule (3)(j).
- (8) Error E01907: Inverse-name of CONTROLLED meta-entity is the same as the assigned-access-name of a CONTROLLED meta-entity. See General Rule (3)(k).
- (9) Error E01908: Inverse-name of CONTROLLED meta-entity is the same as the assigned-descriptive-name of a CONTROLLED meta-entity. See General Rule (3)(k).
- (10) Error E01909: Substitute-name of CONTROLLED meta-entity is the same as the assigned-access-name of a CONTROLLED meta-entity. See General Rule (3)(l).
- (11) Error E01910: Substitute-name of CONTROLLED meta-entity is the same as the assigned-descriptive-name of a CONTROLLED meta-entity. See General Rule (3)(l).
- (12) Error E01911: IRD / IRD Schema inconsistency. Entity has no corresponding type. See General Rule (4)(a).
- (13) Error E01912: IRD / IRD Schema inconsistency. Relationship has no corresponding type. See General Rule (4)(a).

- (14) Error E01913: IRD / IRD Schema inconsistency. Entity or Relationship has attribute with no corresponding type. See General Rule (4)(a).
- (15) Error E01914: IRD / IRD Schema inconsistency. Entity or relationship has a group with no corresponding type. See General Rule (4)(a).
- (16) Error E01915: IRD / IRD Schema inconsistency. Attribute has invalid format or value. See General Rules (4)(b) and (4)(o).
- (17) Error E01916: IRD / IRD Schema inconsistency. Attribute-group is not consistent with corresponding type. See General Rule (4)(c).
- (18) Error E01917: IRD / IRD Schema inconsistency. Corresponding type for attribute not associated with type of corresponding entity. See General Rule (4)(d).
- (19) Error E01918: IRD / IRD Schema inconsistency. Corresponding type for attribute-group not associated with type of corresponding entity. See General Rule (4)(e).
- (20) Error E01919: IRD / IRD Schema inconsistency. Corresponding type for attribute not associated with type of corresponding relationship. See General Rule (4)(f).
- (21) Error E01920: IRD / IRD Schema inconsistency. Corresponding type for attribute-group not associated with type of corresponding relationship. See General Rule (4)(g).
- (22) Error E01921: IRD / IRD Schema inconsistency. Attribute or group repeats too many times. See General Rule (4)(h).
- (23) Error E01922: IRD / IRD Schema inconsistency. Duplicate keys in repeating attribute-group. See General Rule (4)(i).
- (24) Error E01923: IRD / IRD Schema inconsistency. Incomplete key in attribute-group. See General Rule (4)(j).
- (25) Error E01924: IRD / IRD Schema inconsistency. Duplicate sequence attribute. See General Rule (4)(k).
- (26) Error E01925: IRD / IRD Schema inconsistency. Duplicate relationship keys. See General Rule (4)(l).
- (27) Error E01926: IRD / IRD Schema inconsistency. Entity has invalid variation-name. See General Rule (4)(m).
- (28) Error E01927: IRD / IRD Schema inconsistency. Entity has invalid assigned-access-name. See General Rule (4)(n).
- (29) Error E01928: IRD / IRD Schema inconsistency. Entity has invalid assigned-descriptive-name. See General Rule (4)(n).

NOTES:

- (1) An implementation may perform the validation specified in General Rules (3) and (4) only to those descriptors affected by changes to the IRD Schema.
- (2) Action (2)(a) states that when a meta-entity is superseded by another, the superseding meta-entity takes on the

superseded meta-entity's value for the NUMBER-OF-INSTANCES meta-attribute. It also states that any IRD descriptor can be an instance only of a type which is defined in the CONTROLLED IRD Schema life-cycle-phase.

### 5.1.2 IRD Schema Output Command

**Function** To display the contents of the IRD Schema.

**Format**

```

IRD-schema-output-command ::=
    output-word                4.3
    IRD-schema-word            4.3
    [ using-IRD-schema-views-clause ] 6.132
    meta-entity-selection-clause 6.118
    [ sort-meta-entities-clause ] 6.113
    show-IRD-schema-options
    [ route-clause ]          6.64
    ;

show-IRD-schema-options ::=
    [ show-title-clause ]      6.65
    IRD-schema-show-clause-list

IRD-schema-show-clause-list ::=
    IRD-schema-show-clause-option
    [ IRD-schema-show-clause-list ]

IRD-schema-show-clause-option ::=
    IRD-schema-show-all-clause 6.114
    | show-meta-attributes-clause 6.115
    | show-meta-relationships-clause 6.116
    | show-related-meta-entities-clause 6.117
    | show-IRD-schema-life-cycle-phase-clause 6.130

```

#### Syntax Rules

- (1) If IRD-schema-show-all-clause is specified, the show-meta-attributes-clause and show-meta-relationships-clause shall be ignored if specified.

#### General Rules

- (1) If show-meta-attributes-clause specifies a meta-attribute-type which is not associated with the meta-entity-type of a selected meta-entity, the specified meta-attribute-type shall be ignored for that meta-entity. In this case, a warning message shall be produced.
- (2) If no sort-meta-entities-clause is specified, meta-entities shall be displayed in the order selected.
- (3) If using-IRD-schema-views-clause is not specified, then:
  - (a) The only meta-entities which can be selected and displayed shall be those in the IRD Schema life-cycle-phase which corresponds to the effective IRD-schema-view.

- (b) The only meta-relationships which can be displayed shall be those which associate meta-entities in the IRD Schema life-cycle-phase corresponding to the effective IRD-schema-view.
- (4) If using-IRD-schema-views-clause is specified, then:
  - (a) The only meta-entities which can be selected and displayed shall be those in an IRD Schema life-cycle-phase which corresponds to at least one of the specified IRD-schema-views.
  - (b) The only meta-relationships which can be displayed shall be those which associate meta-entities which are in an IRD Schema life-cycle-phase corresponding to at least one of the specified IRD-schema-views.

**Security Rules** None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be produced.
- (2) If no errors are encountered, the meta-entities shall be selected. Additional information is obtained as required by the specified show-options. Information for each meta-entity shall be formatted according to show-options, in the order specified in sort-meta-entities-clause.
- (3) Completion of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Warning W01038: Meta-attribute-type not associated with meta-entity-type or meta-relationship-type.

**5.2 IRD Commands**

**Function** To identify general classes of commands which operate against the IRD and the rules which apply to commands in these classes.

**Format**

/\* None. This breakdown is explanatory only. \*/

```
IRD-command ::=
    IRD-maintenance-command          5.2.1
  | IRD-output-command              5.2.2
```

**Syntax Rules** None.

**General Rules**

- (1) The effective IRDS-user shall have at least one associated IRD-VIEW entity. This is defined in the IRD by an entity of type IRD-VIEW which is related to the effective IRDS-user by a relationship of type IRDS-USER-HAS-IRD-VIEW.

- (2) An IRD-VIEW enables the effective IRDS-user to access and update entities in a given IRD-partition. This correspondence between an IRD-VIEW and an IRD-partition is defined by the attribute IRD-PARTITION-NAME. Each IRD-View shall be associated with a single IRD-partition. Every entity within the IRD-View shall be within the associated IRD-partition. The IRD-PARTITION-NAME associated with the IRD-View shall be the name of an IRD-partition in the IRD Schema.
- (3) The effective IRDS-user can have multiple IRD-VIEWS. For updating the IRD, only one of the IRD-VIEW entities shall be in effect at one time. This is called the effective IRD-view. A default effective IRD-view is determined by the relationship of type IRDS-USER-HAS-IRD-VIEW. For any particular IRDS-USER entity, one and only one relationship of this type shall have the DEFAULT-VIEW attribute equal to YES. The IRD-VIEW entity in this relationship is the default effective IRD-View.
- (4) The IRDS-user can change the effective IRD-view by using the set-session-defaults command (Section 5.3.2.1).

**Security Rules**     None.

**Actions**

- (1) Upon entering the IRDS, the effective IRDS-user shall be assigned a default IRD-View provided one has been defined.

**Error and Warning Conditions**

- (1) Error E01125: IRD-View does not identify a valid IRD-Partition. General Rule (2) of this Subsection has been violated.

**5.2.1 IRD Maintenance Commands**

**Function**     Identify all commands which update the contents of the IRD and the rules which apply to all such commands.

**Format**

/\* None. This breakdown is explanatory only. \*/

IRD-maintenance-command ::=	
add-entity-command	5.2.1.1
modify-entity-command	5.2.1.2
delete-entity-command	5.2.1.3
add-relationship-command	5.2.1.4
modify-relationship-command	5.2.1.5
delete-relationship-command	5.2.1.6
modify-entity-access-name-command	5.2.1.7
modify-entity-descriptive-name-command	5.2.1.8
modify-entity-life-cycle-phase-command	5.2.1.9
copy-entity-command	5.2.1.10

**Syntax Rules**     None.

### General Rules

- (1) Execution of any IRD-maintenance-command requires that the IRD be in the "activated" state.

### Access Rules

For the following Access Rules:

Let  $U$  denote the effective IRDS-user.

Let  $V$  denote the effective IRD-view.

Let  $V_1 \dots V_n$  denote IRD-views which are related to  $U$ .

- (1) In the add-entity-command, copy-entity-command, and modify-entity-command (with new-version-clause specified), when an entity-type being maintained is not in a non-life-cycle-phase IRD-partition,  $V$  shall have the attribute IRD-PARTITION-NAME equal to the assigned-access-name of an IRD-partition meta-entity with meta-attribute LIFE-CYCLE-PHASE-CLASS equal to UNCONTROLLED. This is not true, however, when the entity-type being maintained is in a non-life-cycle-phase IRD-partition.
- (2) For the delete-entity-command and the modify-entity-command (without a new-version-clause specified),  $V$  shall have an IRD-PARTITION-NAME attribute equal to the assigned-access-name of the IRD-partition which contains the specified entity.
- (3) For the add-relationship, modify-relationship, and delete-relationship commands:
  - (a)  $V$  shall have an IRD-PARTITION-NAME attribute equal to the assigned-access-name of the IRD-partition which contains the first entity of the specified relationship.
  - (b) There shall be a  $V_i$ ,  $1 \leq i \leq n$ , with an IRD-PARTITION-NAME attribute equal to the assigned-access-name of the IRD-partition which contains the second meta-entity of the meta-relationship.
- (4) For the modify-entity-life-cycle-phase-command:
  - (a)  $V$  shall have an IRD-PARTITION-NAME attribute equal to the assigned-access-name of the new IRD life-cycle-phase.
  - (b) There shall be a  $V_i$ ,  $1 \leq i \leq n$ , with a PARTITION-NAME attribute equal to the assigned-access-name of the current IRD life-cycle-phase (which contains the entities specified).

**Security Rules** None.

**Actions** None.

### Error and Warning Conditions

- (1) Error E01115: Effective IRD-view not associated with correct IRD-partition. See Access Rules for this Subsection.



### 5.2.1.1 Add Entity Command

**Function** This command shall be used to add a new entity to the IRD and establish initial values for specified attributes.

#### Format

```

add-entity-command ::=
    add-word                4.3
    entity-word             4.3
    new-entity-name
    entity-type-clause      6.2
    [ entity-descriptive-name-declaration-clause ] 6.27
    [ quality-indicator-designation-clause ]       6.30
    [ new-entity-attributes-clause ]              6.22
    ;

new-entity-name ::=
    new-entity-access-name 4.4
    | null-mark

```

#### Syntax Rules

- (1) If new-entity-access-name is specified:
  - (a) The assigned-access-name of the entity shall conform to the length and format rules as defined in the IRD Schema for the entity-type identified in the entity-type-clause.
  - (b) It shall not be equal to any potential system-generated name. (System-generated assigned-access-names are of the form *PN*, where *P* is a prefix composed of non-numeric naming characters and *N* is an integer within the range specified for the entity-type. The naming convention is assigned on an entity-type by entity-type basis.)
- (2) The version-identifier shall not specify a revision-number that already exists for the specified assigned-access-name and variation-name combination.
- (3) The entity-type-clause shall immediately follow new-entity-name.
- (4) If specified, the entity-descriptive-name-declaration-clause shall immediately follow entity-type-clause.
- (5) All attribute declarations shall be part of the new-entity-attributes-clause.
- (6) If the entity-type identified in the entity-type-clause is defined in the IRD Schema with a value of YES for the meta-attribute-type SYSTEM-GENERATED, then either null-mark or new-entity-name can be specified.
- (7) New-entity-access-name shall be specified if the entity-type identified in entity-type-clause is defined in the IRD Schema with a value of NO for the meta-attribute SYSTEM-GENERATED.
- (8) The entity-type identified in entity-type-clause shall be defined in the IRD Schema, in the CONTROLLED IRD Schema life-cycle-phase.

- (9) If the entity-type identified in entity-type-clause is either IRDS-USER, IRD-VIEW, or IRD-SCHEMA-VIEW, then no version-identifier shall be specified in new-entity-access-name.
- (10) If specified, quality-indicator-designation-clause shall precede new-entity-attributes-clause.

**General Rules**

- (1) If a revision-number is not explicitly specified, the add-entity-command shall always create an entity with a revision-number equal to one.
- (2) If the entity-type specified is IRDS-USER, IRD-VIEW, or IRD-SCHEMA-VIEW, then the IRD-VIEW which is in effect shall be associated with the IRD-PARTITION named SECURITY.
- (3) If the IRD-VIEW which is in effect is associated with the IRD-PARTITION named SECURITY, then only entities of the types IRDS-USER, IRD-VIEW, or IRD-SCHEMA-VIEW shall be added.

**Security Rules** None.

**Actions**

- (1) If the command passes validation, then:
  - (a) The entity shall be created in the IRD.
  - (b) The created entity shall be within the IRD-Partition which is associated with the IRD-VIEW in effect.
  - (c) This command shall assign appropriate values to the audit-attribute ADDED-BY and the audit-attribute-group DATE-TIME-ADDED. It shall treat the audit-attribute-group DATE-TIME-LAST-MODIFIED and the audit-attribute LAST-MODIFIED-BY as non-existent, and the NUMBER-OF-TIMES-MODIFIED audit-attribute shall be assigned a value of zero.
  - (d) If the entity has a system-generated-name and new-entity-name is not specified, then
    - (i) The generated entity-access name assigned to the entity shall be  $PN$ , where
      - (A)  $P$  is the prefix in the start-name for the entity-type;
      - (B)  $PN$  is not the assigned-access-name of an existing entity; and
      - (C)  $N$  is an integer such that if the START-NAME for the entity-type is  $Pk_1k_2 \dots k_j$ ,  $N$  falls in the range  $L \leq N \leq U$ , where
 
$$L = (k_1) \cdot (10^{j-1}) + \dots + k_j, \text{ and}$$

$$U = 9 \cdot (10^{j-1}) + \dots + 9.$$
    - (ii) The system shall inform the user of the generated entity-access-name.
    - (iii) The entity shall have the attribute of type SYSTEM-GENERATED-NAME equal to YES.

- (e) If new-entity-name is specified, the entity shall have the attribute of type SYSTEM-GENERATED-NAME equal to NO.
  - (f) If quality-indicator-designation-clause is specified, the entity shall be associated with the specified quality-indicator.
- (2) If the command fails to pass validation, the entity shall not be created in the IRD. The user shall be informed that the entity has not been created, and shall be given all appropriate error messages.

**Error and Warning Conditions**

- (1) Error E01002: Invalid or unrecognized entity-type. Either Syntax Rule (3) or (8) for this command has been violated.
- (2) Error E01004: User designated entity-access-name is required. Syntax Rule (7) for this command has been violated.
- (3) Error E01005: Invalid name format. Syntax Rule (1) for this command has been violated.
- (4) Error E01006: Invalid version-identifier. Either Syntax Rule (2) for this command has been violated, or the version-identifier format is incorrect.
- (5) Error E01115: Effective IRD-view not associated with correct IRD-partition. General Rule (2) or (3) for this command has been violated.
- (6) Error E01022: Multiple versions not allowed for entities of entity-type. Syntax Rule (9) for this command has been violated.

**5.2.1.2 Modify Entity Command**

**Function** To modify the attributes of an existing entity in the IRD or to create a new entity with the same assigned-access-name and different version-identifier as an existing entity.

**Format**

```

modify-entity-command ::=
    modify-word           4.3
    entity-word           4.3
    existing-entity-access-name 4.4
    [ entity-type-clause ] 6.2
    [ new-version-clause ] 6.28
    [ entity-descriptive-name-declaration-clause ] 6.27
    [ quality-indicator-designation-clause ] 6.30
    [ modified-entity-attributes-clause ] 6.23
    ;
    
```

**Syntax Rules** None.

**General Rules**

- (1) The entity-type-clause within this command shall be informational only. However, if

specified, the entity-type-clause shall identify the entity-type of the entity identified by existing-entity-access-name.

- (2) If the new-version-clause is specified:
  - (a) The assigned-access-name of the existing entity and the version-identifier specified by the new-version-clause shall not identify an existing entity within the IRD.
  - (b) If a version-identifier is specified with a variation-name, the variation-name shall be valid for the entity-type.
  - (c) If a revision-number is specified within version-identifier, it shall be different than any other revision-number of any entity with the specified assigned-access-name and the specified or defaulted variation-name.
- (3) New-version-clause shall not be specified if the entity-type of the specified entity is IRDS-USER, IRD-VIEW, or IRD-SCHEMA-VIEW.
- (4) Entity-descriptive-name-declaration-clause shall be allowed only if the entity being modified does not have a descriptive-name.

**Security Rules**     None.

#### **Actions**

- (1) If all command-clauses pass validation, and a new-version-clause has not been specified, the attributes of the existing entity shall be modified as specified within modified-entity-attributes-clause. If quality-indicator-designation-clause is specified, the modified entity shall be associated with the specified quality-indicator. The system shall confirm this action for the user. This command creates or updates the audit-attribute-group DATE-TIME-LAST-MODIFIED and the audit-attribute LAST-MODIFIED-BY for the maintained entity. The value of the NUMBER-OF-TIMES-MODIFIED audit-attribute shall be incremented by one. If a valid entity-descriptive-name-clause is specified, then all entities with the same assigned-access-name as the specified entity shall be given the descriptive-name.
- (2) If all command-clauses pass validation, and a new-version-clause has been specified, then the existing entity shall remain unaffected. A new entity, whose entity-access-name has the same assigned-access-name as the existing entity, but whose version-identifier is as specified in the new-version-clause, shall be created. If quality-indicator-designation-clause is specified, the new entity shall be associated with the specified quality-indicator.

The attributes of the new entity shall be the same as those of the existing entity, except for those modifications specified in the modified-entity-attributes-clause. The following shall have new values: the audit-attribute-group DATE-TIME-ADDED and the audit-attribute ADDED-BY. The audit-attribute-group DATE-TIME-LAST-MODIFIED and the audit-attribute LAST-MODIFIED-BY shall be treated as nonexistent. The NUMBER-OF-TIMES-MODIFIED audit-attribute shall be assigned a value of zero.

If existing-entity-name has an entity-descriptive-name, a new entity-descriptive-name shall be constructed for the new entity. This entity-descriptive-name shall have the same assigned-descriptive-name as that of existing-entity-access-name, and the version-identifier assigned to the new entity.

If existing-entity-access-name does not have an entity-descriptive-name and entity-descriptive-name-declaration-clause is specified, then all entities with the same assigned-access-name shall be assigned a descriptive-name.

Relationships which contain the new entity shall be derived from relationships containing the specified entity. New relationships shall be created according to the following rules:

- (a) For each relationship which relates the specified entity to itself, (i.e., the specified entity is both the first and second entity of the relationship) a new relationship of the same type shall be created which relates the new entity to itself. The attributes of corresponding relationships shall be identical.
  - (b) For each relationship which has the specified entity only as the first entity, a new relationship of the same type shall be created with the new entity as the first entity. The second entity and the attributes of the corresponding relationships shall be identical.
  - (c) For each relationship which has the specified entity only as the second entity, a new relationship of the same type with the new entity as the second entity shall be created only if the relationship-type has the SEQUENCED meta-attribute equal to NO. The first entity and the attributes of corresponding relationships shall be identical.
- (3) If any command-clause is not valid, the user shall be given all appropriate error messages and informed that the command has been rejected.

#### **Error and Warning Conditions**

- (1) Error E01010: Entity is not of specified entity-type. General Rule (1) of this Subsection is violated. Possibly the entity-type specified in the entity-type-clause is not valid.
- (2) Error E01022: Multiple versions not allowed for entities of entity-type. General Rule (3) for this command has been violated.
- (3) Warning W01008: One or more relationships have not been duplicated for the new-version of the entity modified.
- (4) Error E01059: Descriptive-name may not be changed by a modify-entity command. General Rule (4) for this command has been violated.

#### **5.2.1.3 Delete Entity Command**

**Function** To delete one or more entities from the IRD.

#### **Format**

```

delete-entity-command ::=
    delete-word           4.3
    entity-word           4.3
    delete-selection-option
    [ with-relationships-clause ] 6.32
    ;

```

```
delete-selection-option ::=
    entity-name-list
    | entity-selection-criteria-clause      6.33

entity-name-list ::=
    entity-access-name                      4.4
    [ , entity-name-list ]
```

**Syntax Rules** None.

### General Rules

- (1) Each entity-access-name specified directly in the command shall identify an entity which exists in the IRD.
- (2) Unless the with-relationships-clause is specified, each entity to be deleted shall not participate in any relationship.

**Security Rules** None.

### Actions

- (1) The command shall process according to the rules stated in the introduction to Section 5.
- (2) If with-relationships-clause is specified, every relationship in which each specified entity participates will also be deleted from the IRD.
- (3) The user shall be notified of completion of the delete-entity-command.

### Error and Warning Conditions

- (1) Error E01009: Entity does not exist or is not visible. General Rule (1) for this command has been violated.
- (2) Error E01013: Entity is related to other entities. General Rule (2) for this command has been violated.

#### 5.2.1.4 Add Relationship Command

**Function** To establish a relationship between two entities in the IRD.

### Format

```
add-relationship-command ::=
    add-word                               4.3
    relationship-word                       4.3
    new-relationship-identification-clause  6.25
    [ new-relationship-attributes-clause ]  6.43
    ;
```

**Syntax Rules** None.

**General Rules**

- (1) The relationship specified in new-relationship-identification-clause shall not exist within the IRD.

**Security Rules** None.

**Actions**

- (1) If the command passes validation then:
- (a) If the new-relationship-identification-clause specifies new-word prior to entity-2, then an entity with the specified entity-access-name shall be added to the IRD. The new entity shall be within the IRD-Partition associated with the IRD-VIEW which is in effect.
  - (b) The relationship shall be created and added to the IRD.
  - (c) The system shall confirm completion of the action.
- (2) If the command does not pass validation, the command shall be rejected. The system shall provide all appropriate error messages, and inform the user that the command has been rejected.

**Error and Warning Conditions**

- (1) Error E01017: Relationship already exists. General Rule (1) for this command has been violated.

**5.2.1.5 Modify Relationship Command**

**Function** To modify the attributes of an existing relationship in the IRD.

**Format**

```

modify-relationship-command ::=
    modify-word                4.3
    relationship-word          4.3
    relationship-identification-clause 6.24
    [ modified-relationship-attributes-clause ] 6.44
    ;

```

**Syntax Rules** None.

**General Rules**

- (1) The relationship specified in relationship-identification-clause shall exist within the IRD.

**Security Rules** None.

**Actions**

- (1) If the command passes validation then the attributes of the relationship shall be modified

as specified in the modified-relationship-attributes-clause (See Subsection 6.44). The system shall provide confirmation of completion of the modification.

- (2) If the command does not pass validation, the command shall be rejected. The system shall provide all appropriate error messages, and inform the user that the command has been rejected.

#### **Error and Warning Conditions**

- (1) Error E01018: Relationship does not exist. General Rule (1) for this command has been violated.

#### **5.2.1.6 Delete Relationship Command**

**Function** To delete one or more relationships from the IRD.

#### **Format**

```
delete-relationship-command ::=
    delete-word                4.3
    relationship-word           4.3
    relationships-identification
    ;

relationships-identification ::=
    relationship-selection-clause 6.45
    | relationship-identification-list

relationship-identification-list ::=
    relationship-identification-clause 6.24
    [ [,] relationship-identification-list ]
```

**Syntax Rules** None.

#### **General Rules**

- (1) Each relationship specified in a relationship-identification-clause shall exist within the IRD.

**Security Rules** None.

#### **Actions**

- (1) The command shall be processed according to the General Rules in the introduction to Section 5.

#### **Error and Warning Conditions**

- (1) Error E01018: Relationship does not exist. General Rule (1) for this command has been violated.
- (2) Warning W01006: No relationships obtained. No relationships existed which fulfilled the selection criteria of the relationship-selection-clause.



- (3) **Warning W01004:** Command executed with errors. Either at least one relationship specified within the relationship-identification-list could not be deleted, or execution errors were encountered in attempting to select relationships.

**5.2.1.7 Modify Entity-Access-Name Command**

**Function** To change the assigned-access-name of all entities with a given assigned-access-name.

**Format**

```

modify-entity-access-name-command ::=
    modify-word           4.3
    entity-word           4.3
    access-name-word      4.3
    from-word             4.3
    existing-entity-assigned-access-name
    to-word               4.3
    new-entity-assigned-access-name
    ;

existing-entity-assigned-access-name ::=
    assigned-access-name  4.4

new-entity-assigned-access-name ::=
    assigned-access-name  4.4
    
```

**Syntax Rules** None.

**General Rules**

- (1) Existing-entity-assigned-access-name shall be the assigned-access-name of at least one existing entity in the IRD.
- (2) New-assigned-access-name shall not be the assigned-access-name of any entity in the IRD, nor shall it be equal to the assigned-descriptive-name of any entity in the IRD.
- (3) New-assigned-access-name shall conform to all the rules for naming an entity of the entity-type of current-assigned-access-name.
- (4) The value of new-assigned-access-name shall not be equal to any potential system-generated access-name.

**Security Rules** None.

**Actions**

- (1) If the command passes validation, then:
  - (a) Each entity with an assigned-access-name of existing-entity-assigned-access-name shall be changed to have an assigned-access-name of new-assigned-access-name.
  - (b) All version-identifiers shall remain unchanged.

- (c) New values shall be assigned to the audit-attribute-group DATE-TIME-LAST-MODIFIED and the audit-attribute LAST-MODIFIED-BY. The value of the NUMBER-OF-TIMES-MODIFIED audit-attribute shall be incremented by one.
- (2) If an entity whose assigned-access-name is changed has the attribute of type SYSTEM-GENERATED-NAME equal to YES, this attribute shall be changed to NO.

**Error and Warning Conditions**

- (1) Error E01009: Entity does not exist or is not visible. General Rule (1) for this command has been violated.
- (2) Error E01007: Entity-access-name conflict. General Rule (2) or (4) for this command has been violated.
- (3) Error E01005: Invalid name format. General Rule (3) for this command has been violated.

**5.2.1.8 Modify Entity-Descriptive-Name Command**

**Function** To change the assigned-descriptive-name of all entities with a given assigned-descriptive-name.

**Format**

```

modify-entity-descriptive-name-command ::=
    modify-word           4.3
    entity-word           4.3
    descriptive-name-word 4.3
    existing-entity-specification
    to-word               4.3
    new-entity-assigned-descriptive-name
    ;

existing-entity-specification ::=
    using-entity-assigned-descriptive-name-
        specification
    | using-entity-assigned-access-name-
        specification

using-entity-assigned-descriptive-name-
specification ::=
    from-word           4.3
    existing-entity-assigned-
        descriptive-name

using-entity-assigned-access-name-
specification ::=
    for-word           4.3
    existing-entity-assigned-access-name

existing-entity-assigned-descriptive-name ::=
    assigned-descriptive-name           4.4
    
```

existing-entity-assigned-access-name ::=	assigned-access-name	4.4
new-entity-assigned-descriptive-name ::=	assigned-descriptive-name	4.4

**Syntax Rules**    None.

### General Rules

- (1) If using-entity-assigned-descriptive-name-specification is used, then existing-entity-assigned-descriptive-name shall be the assigned-descriptive-name of at least one existing entity in the IRD.

If using-entity-assigned-access-name-specification is used, then existing-entity-assigned-access-name shall be the assigned-access-name of at least one existing entity in the IRD, and the descriptive-name of each such entity shall be null.

- (2) New-entity-assigned-descriptive-name shall not be the same as:
- (a) The assigned-access-name of any existing entity, except for those entities identified by existing-entity-assigned-descriptive-name.
  - (b) The assigned-descriptive-name of an existing entity.
- (3) New-entity-assigned-descriptive-name shall conform to all the rules for an entity assigned-descriptive-name.

**Security Rules**    None.

### Actions

- (1) If the command passes validation, then:
- (a) Each entity with an assigned-descriptive-name of existing-entity-descriptive-name shall be changed to have an assigned-descriptive-name of new-entity-assigned-descriptive-name.
  - (b) Each entity with an assigned-access-name of existing-entity-assigned-access-name shall be changed to have an assigned-descriptive-name of new-entity-assigned-descriptive-name.
  - (c) All version-identifiers shall remain unchanged.
  - (d) New values shall be assigned to the audit-attribute-group DATE-TIME-LAST-MODIFIED and the audit-attribute LAST-MODIFIED-BY. The value of the NUMBER-OF-TIMES-MODIFIED audit-attribute shall be incremented by one.

### Error and Warning Conditions

- (1) Error E01009: Entity does not exist or is not visible. General Rule (1) for this command has been violated.

- (2) Error E01031: Entity-descriptive-name conflict. General Rule (2) for this command has been violated.
- (3) Error E01032: Invalid descriptive-name. General Rule (3) for this command has been violated.

### 5.2.1.9 Modify Entity Life-Cycle-Phase Command

**Function** To transfer one or more entities from one IRD Life-Cycle-Phase to another.

#### Format

```
modify-entity-life-cycle-phase-command ::=
    modify-word           4.3
    entity-word           4.3
    life-cycle-phase-word 4.3
    [ for-word ]         4.3
    entity-name-list
    from-word            4.3
    current-life-cycle-phase
    to-word              4.3
    new-life-cycle-phase
    ;

current-life-cycle-phase ::=
    irds-name           4.2

new-life-cycle-phase ::=
    irds-name           4.2

entity-name-list ::=
    entity-access-name  4.4
    [ , entity-name-list ]
```

#### Syntax Rules

- (1) Both `current-life-cycle-phase` and `new-life-cycle-phase` shall be valid life-cycle-phase-names.
- (2) `Current-life-cycle-phase` and `new-life-cycle-phase` shall be different life-cycle-phase-names.

#### General Rules

- (1) Each `entity-access-name` shall identify an existing entity in the IRD.
- (2) Each `entity-access-name` shall identify an entity which is in `current-life-cycle-phase`.

**Security Rules** None.

#### Actions

- (1) Processing of this command shall be as specified in the introduction to Section 5.

- (2) The modification step shall transfer each entity selected in the prior step from current-life-cycle-phase to new-life-cycle-phase. For each processed entity, the NUMBER-OF-TIMES-MODIFIED audit-attribute shall be incremented by one, and appropriate values shall be assigned to the DATE-TIME-LAST-MODIFIED audit-attribute-group and the LAST-MODIFIED-BY audit-attribute. All relationships in which the transferred entity participates shall remain unaffected.

**Error and Warning Conditions**

- (1) Error E01033: Unrecognized IRD life-cycle-phase-designator. Syntax Rule (1) for this command has been violated.
- (2) Error E01034: Invalid phase transition. Syntax Rule (2) for this command has been violated.
- (3) Error E01009: Entity does not exist or is not visible. General Rule (1) for this command has been violated.
- (4) Error E01035: Entity not in specified phase. General Rule (2) for this command has been violated.

**5.2.1.10 Copy Entity Command**

**Function** To create a new entity with the same attributes (and optionally the relationships to the same entities) as an existing entity.

**Format**

```

copy-entity-command ::=
    copy-word                4.3
    entity-word              4.3
    existing-entity-access-name 4.4
    [ with-relationships-clause ] 6.32
    to-word                  4.3
    new-entity
    [ entity-descriptive-name-declaration-clause ] 6.27
    [ quality-indicator-designation-clause ] 6.30
    ;

new-entity ::=
    new-entity-access-name 4.4
    | null-mark            4.1
    | new-version-clause   6.28
    
```

**Syntax Rules**

- (1) If new-entity-access-name is specified, its assigned-access-name shall not be the same as that of existing-entity-access-name. (To create a new version of the entity, new-version-clause shall be used.)
- (2) Entity-descriptive-name-declaration-clause shall not be specified if new-version-clause is specified for new-entity.

## General Rules

- (1) If the new-version-clause is specified:
  - (a) The assigned-access-name of the existing entity and the version-identifier specified by the new-version-clause shall not identify an existing entity within the IRD.
  - (b) If a version-identifier is specified with a variation-name, the variation-name shall be valid for the entity-type of the entity being copied.
  - (c) If a revision-number is specified within version-identifier, it shall be different than any other revision-number of any entity with the specified assigned-access-name and the specified or defaulted variation-name.
- (2) New-version-clause shall not be specified if the entity-type of existing-entity-access-name is in a non-life-cycle-phase IRD-Partition.
- (3) If the entity-type of existing-entity-access-name is defined in the IRD Schema with a value of YES for the meta-attribute-type SYSTEM-GENERATED, then either null-mark or new-entity-name can be specified.
- (4) The entity-type of new-entity-access-name shall be the same as the entity-type of existing-entity-access-name.
- (5) Entity-descriptive-name-declaration-clause shall be allowed only if the entity being copied does not have a descriptive-name.

**Security Rules**     None.

## Actions

- (1) Once the command has passed validation, a new entity of the specified or defaulted entity-access-name shall be created. The audit-attribute ADDED-BY and the audit-attribute-group DATE-TIME-ADDED shall be assigned appropriate values. The audit-attribute-group DATE-TIME-LAST-MODIFIED and the audit-attribute LAST-MODIFIED-BY shall be treated as non-existent, and NUMBER-OF-MODIFICATIONS shall be assigned a value of zero.
- (2) If new-entity-access-name is specified, and the entity with existing-entity-access-name has the attribute of type SYSTEM-GENERATED-NAME equal to YES, the new entity shall have this attribute equal to NO.
- (3) If new-version-clause is specified, then the new entity, whose entity-access-name has the same assigned-access-name as existing-entity-access-name, but whose version-identifier is as specified in the new-version-clause, shall be created. The new entity shall have the same assigned-descriptive-name as the existing entity.
- (4) If with-relationships-clause is specified, relationships which contain the new entity shall be derived from relationships containing the entity identified by new-entity-access-name. New relationships shall be created according to the following rules:
  - (a) For each relationship which relates the existing entity to itself, (i.e., the existing entity is both the first and second entity of the relationship) a new relationship of

the same type shall be created which relates the new entity to itself. The attributes of corresponding relationships shall be identical.

- (b) For each relationship which has the existing entity only as the first entity, a new relationship of the same type shall be created with the new entity as the first entity. The second entity and the attributes of the corresponding relationships shall be identical.
- (c) For each relationship which has the existing entity only as the second entity, a new relationship of the same type with the new entity as the second entity shall be created only if the relationship-type has the SEQUENCED meta-attribute equal to NO. The first entity and the attributes of corresponding relationships shall be identical.
- (5) If entity-descriptive-name-declaration-clause is specified, then the appropriate descriptive-name shall be declared.
- (6) If quality-indicator-designation-clause is specified, then the entity identified by new-entity-access-name shall be associated with the specified quality-indicator.

**Error and Warning Conditions**

- (1) Error E01004: User designated entity-access-name required. See General Rule (3) for this command. The entity-type of existing-entity-access-name does not have SYSTEM-GENERATED access-names.
- (2) Error E01022: Multiple versions not allowed for entities of entity-type. General Rule (2) for this command has been violated.
- (3) Error E01036: New-entity-access-name cannot be the same as the existing-entity-access-name. Syntax Rule (1) has been violated.
- (4) Error E01037: Descriptive-name cannot be assigned. General Rule (5) has been violated.

**5.2.2 IRD Output Commands**

**Function** To query the IRD.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

IRD-output-commands ::=
    general-output-command           5.2.2.1
    | output-impact-of-change-command 5.2.2.2
    | [ output-syntax-command ]      5.2.2.3
    
```

**Syntax Rules**

- (1) The general form of these commands is as follows:

```

command-imperative
    [ using-IRD-views-clause ]      6.38
    selection-option
    
```

[ sort-clause ]	6.62
show-options	
[ route-clause ]	6.64
[ implementor-defined-clauses ]	
;	

where command-imperative is a unique sequence of irds-words which identify the particular command.

- (2) Selection-option is always defined as follows:

selection-option ::= entity-selection-criteria-clause

- (3) Show-options are defined on a command by command basis. In general, show-options consist of one or more show-clauses.
- (4) Show-clauses are of two general classes, qualified and unqualified. Unqualified show-clauses define output formatting regardless of what entity-type or relationship-type shall be being displayed. Qualified show-clauses define what shall be displayed for a particular entity-type or relationship-type.
- (5) All show-clauses shall be specified consecutively.
- (6) All unqualified show-clauses shall be specified before all qualified show-clauses.
- (7) Show-clauses have different levels of control. The first level is the global level, which contains all unqualified show-clauses. The second level is the entity-type level, which contains all show-clauses which pertain to displaying information on entities of a particular entity-type. The third level is the relationship-type level. This contains all show-clauses which pertain to displaying information on relationships of a particular type.
- (8) Show-clauses shall be coded hierarchically.

### General Rules

- (1) The output-syntax-command shall be necessary and meaningful only in the event that the Command Language Interface is available.
- (2) If specified, using-IRD-views-clause shall restrict the selection of entities which shall be included in the entity-list. No entity within the list shall be selected unless it is visible in at least one of the specified IRD-Views.
- (3) If no using-IRD-views-clause is specified within this command, the IRD-View in effect shall be assumed.
- (4) If no sort-clause is specified and multiple entities are selected, the order of retrieval shall be assumed.
- (5) It shall be possible to sort entities based upon attributes and/or names which are not specified in the show-clause.
- (6) Show-clauses also implicitly specify selection of relationships and attributes.



- (7) For entities, qualified show-clauses depend upon the entities selected according to the selection option. In general, no error condition shall be raised if a show-clause is qualified by an entity-type which is not directly or indirectly specified in the selection criteria. The exception to this rule is if the command specifies some predefined display which does not show entities of the entity-type specified as a qualifier.
- (8) A qualified show-clause which describes how to display a given relationship-type depends on show-clauses at a higher level of qualification, (including unqualified show-clauses). It shall not be possible to use a relationship-type as a qualifier for a show-clause if either the command does not support the display of relationships of the type used as a qualifier, or the selection of such relationships was suppressed by a clause at a higher level of control.
- (9) A show-clause at a lower level of qualification shall take precedence over a show-clause at a higher level of qualification. Thus, if an unqualified show-clause specifies "show all attributes" and a qualified show-clause specifies "for entity-type *x* show no attributes", no attributes shall be displayed for entities of entity-type *x*.
- (10) The default for routing output shall be implementation dependent. It is suggested that this be a session option.
- (11) Implementor-defined command-clauses may be used to define any implementation-dependent processing options.

**Security Rules**     None.

#### **Actions**

- (1) The command shall be validated. If any Syntax Rule or General Rule is violated, the command does not execute. All error and warning conditions shall be given. The command shall be then executed.
- (2) Entities shall be selected. Selection shall be done directly from the IRD if entity-selection-criteria-clause was specified. If using-list-clause is specified, entities shall be selected from the identified entity-list, provided they are visible within the IRD-Views used by the command. If sort-clause is specified, any additional IRD information used to sort the entities shall also be obtained.

If sort-clause is specified, all selected information shall be sorted as specified in the sort-clause.

Additional information is selected from the IRD to format the specified output. (The implementation may choose to perform this selection either prior to or after the sort. The choice may vary by the type of display. This standard does not specify or imply any implementation details. The implementation may perform these processes in as many or few phases as is deemed desirable.)

The information shall be routed to the appropriate destination and formatted according to the specifications of the show-clause(s).

- (3) The system shall confirm completion of the command.

## Error and Warning Conditions

- (1) Error E01044: Invalid, missing, or misplaced command-clause. See Syntax Rules (5) through (8).
- (2) Error E01040: Inconsistent command-clauses. See General Rules (7) and (8).

### 5.2.2.1 General Output Command

**Function** To retrieve IRD descriptors, format output, and route the output to specified destinations.

#### Format

```

general-output-command ::=
    output-word                4.3
    IRD-word                   4.3
    [ using-IRD-views-clause ] 6.38
    selection-option
    [ sort-clause ]           6.62
    show-options
    [ route-clause ]         6.64
    [ implementor-defined-clauses ]
    ;

selection-option ::=
    entity-selection-criteria-clause 6.33

show-options ::=
    | show-predefined-display-clause 6.63
    | show-clause-list

show-clause-list ::=
    [ show-title-clause ]           6.65
    unqualified-show-clause-list
    [ qualified-show-clause-list

unqualified-show-clause-list ::=
    general-output-show-clause-option
    [ unqualified-show-clause-list ]

qualified-show-clause-list ::=
    qualified-show-clause           6.67
    [ qualified-show-clause-list ]

general-output-show-clause-option ::=
    show-all-clause               6.74
    | show-entity-type-clause      6.71
    | show-IRD-life-cycle-phase-clause 6.68
    | show-entity-access-name-clause 6.69
    | show-entity-descriptive-name-clause 6.70
    | show-relationships-clause    6.72
    | show-attributes-clause       6.73

```

show-counts-clause	6.82
show-quality-indicator-clause	6.83

```
implementor-defined-clauses ::=
    /* implementor-defined-format */
```

### Syntax Rules

- (1) Within unqualified-show-clause-list, no command-clause shall be repeated.
- (2) At any level of qualification (including no qualification) if show-all-clause is specified, no other show-clause shall be specified.
- (3) No command-clause of the type identified in the general-output-show-clause-option shall be repeated for the same qualifiers.

### General Rules

- (1) It is possible for a predefined-display to have an implicit default sort. In this case, the displayed order of entities may not be retrieval order.
- (2) It is possible that a sort-clause may not be allowed for a specific predefined display. In this case, a warning condition shall be raised and sort-clause is ignored.

**Security Rules** None.

### Actions

- (1) As specified in Actions of "IRD Output Commands", Subsection 5.2.2.

### Error and Warning Conditions

- (1) Error E01108: Duplicated command-clause. See Syntax Rules (1) and (3).
- (2) Error E01040: Inconsistent command-clauses. See Syntax Rule (2).
- (3) Warning W01014: Sort-clause ignored. See General Rule (2).

#### 5.2.2.2 Output Impact-of-Change Command

**Function** To identify those entities which are potentially impacted by changes to selected entities.

### Format

```
output-impact-of-change-command ::=
    output-word           4.3
    [ impact-option ]
    impact-word           4.3
    [ using-IRD-views-clause ] 6.38
    selection-option
    [ sort-clause ]      6.62
    [ impact-show-options ]
```

[ route-clause ]	6.64
[ implementor-defined-clauses ]	
;	
impact-option ::=	
cumulative-word	4.3
individual-word	4.3
selection-option ::=	
entity-selection-criteria-clause	6.33
impact-show-options ::=	
[ show-title-clause ]	6.65
[ entity-type-show-restriction-clause ]	6.66
{ impacted-entity-show-clause }	
{ qualified-impacted-entity-show-clause }	
impacted-entity-show-clause ::=	
show-IRD-life-cycle-phase-clause	6.68
show-quality-indicator-clause	6.83
show-attributes-clause	6.73
show-entity-descriptive-name-clause	6.70
qualified-impacted-entity-show-clause ::=	
entity-type-qualification-clause	6.1
impacted-entity-show-clause-list	
impacted-entity-show-clause-list ::=	
impacted-entity-show-clause	
[ impacted-entity-show-clause-list ]	

### Syntax Rules

- (1) If impact-option is not specified, individual-word shall be assumed.
- (2) If cumulative-word is specified for impact-option, sort-clause shall be ignored. If specified, a warning condition shall be raised.
- (3) The entity-types identified in different entity-type-qualification-clauses shall be unique.
- (4) Each impacted-entity-show-clause shall be specified only once per unique qualification (including the unqualified command-clauses).
- (5) If entity-type-show-restriction-clause is specified, each entity-type specified in an entity-type-qualification-clause shall also be specified in the entity-type-show-restriction-clause.
- (6) Each attribute-type specified in a qualified show-attributes-clause shall be associated with each entity-type in the corresponding entity-type-qualification-clause.

### General Rules

- (1) Let  $E(i) R E(j)$  denote a relationship of type  $R$  in which  $E(i)$  is entity-1 and  $E(j)$  is entity-2 of the relationship.

For any given entity selected  $E(s)$ , the set of entities potentially impacted by a change in  $E(s)$  is to be defined to be the set of all entities which are denoted by  $E'$  such that:  $E' R E(s)$  for some relationship-type  $R$ .

- (2) If impact-option is specified or assumed to be individual-word, then this command shall produce a list of potentially impacted entities for each entity selected according to selection-option. In this case, there shall be a heading which clearly associates each list with the corresponding selected entity.
- (3) If impact-option is cumulative-word, then only one list shall be produced which identifies the set of all entities which are potentially impacted by any of the selected entities. This is equivalent to taking the union of all the individual impacted entity sets. (This does not imply an implementation for this option.)
- (4) The set of potentially impacted entities shall be displayed in sequence by entity-type and entity-access-name. The display shall clearly identify this grouping by entity-type.
- (5) Entity-type-show-restriction-clause enables the list of potentially impacted entities to be restricted to specified entity-types. If this command-clause is specified, then an entity shall be displayed only if its corresponding entity-type is specified in this command-clause.
- (6) If sort-clause is specified and impact-option is specified or assumed as individual-word, then each individual list of impacted entities shall be displayed according to the order of selected entities.
- (7) The qualified and unqualified impacted-entity-show-clauses shall be used to specify which additional information to be displayed for each potentially impacted entity.

**Security Rules**     None.

#### **Actions**

As specified in Actions of "IRD Output Commands", Subsection 5.2.2.

#### **Error and Warning Conditions**

- (1) Warning W01014: Sort-clause ignored. See Syntax Rule (2).
- (2) Error E01107: Duplicate qualification. Syntax Rule (3) has been violated.
- (3) Error E01108: Duplicated command-clause. Syntax Rule (4) has been violated.
- (4) Error E01109: Entities of specified entity-type will not be shown. Syntax Rule (5) has been violated.
- (5) Error E01040: Inconsistent command-clauses. Syntax Rule (6) has been violated.

#### **5.2.2.3 Output Syntax Command**

**Function**     To display selected contents of the IRD as IRD command language syntax.

**Format**

output-syntax-command ::	
output-word	4.3
syntax-word	4.3
[ using-IRD-views-clause ]	6.38
selection-option	
[ sort-clause ]	6.62
[ syntax-show-options ]	
[ route-clause ]	6.64
[ implementor-defined-clauses ]	
;	
selection-option ::=	
entity-selection-criteria-clause	6.33
syntax-show-options ::=	
[ show-title-clause ]	6.65
[ show-IRD-life-cycle-phase-clause ]	6.68
[ show-relationships-clause ]	6.72
[ show-relationship-syntax-clause ]	6.75

**Syntax Rules**

- (1) The command-clauses within syntax-show-options can be specified in any order. Each such command-clause shall be specified at most once.

**General Rules**

- (1) Syntax presentation shall be consistent with but not identical to the coding of add-entity-command and the add-relationship-command. Details of presentation follow:

- (a) The syntax presentation for each entity shall start with:

```
BEGIN ENTITY entity-access-name
      ENTITY-TYPE = entity-type-name
```

More than one line can be used.

- (b) Relationship syntax presentation shall start with:

```
BEGIN relationship-identification-clause
```

Relationship-identification-clause has two forms. In one form, the relationship-type-designator is specified between the entity-access-names of the two related entities. In the other form, the relationship-class-type-designator is used. Where the relationship is of a type which has an associated relationship-class-type, the preferable form of relationship-identification-clause is the one which uses the relationship-class-type-designator.

- (c) All user-specifiable attributes, and attribute-groups are shown as if they were part of an add-entity-command or an add-relationship-command. In particular, all attributes shall appear after with-word, and text attributes are shown as:

text-attribute-type-designator =  
 string-literal-1 ..., string-literal-n

where there is 1 string-literal per line of text.

- (d) Only the meta-entity-access-names of entity-types, relationship-types, relationship-class-types, attribute-types and attribute-group-types shall be displayed in this output display.
  - (e) For each entity and relationship, all attributes and attribute-groups shall be presented alphabetically by attribute-type-designator or attribute-group-type-designator.
  - (f) Repeating attribute-groups shall be presented in order of the values of the significant component attributes.
  - (g) A semicolon shall appear after the last attribute or attribute-group for the entity or relationship.
- (2) If no sort-clause is specified, the syntax for entities shall be sequenced by entity-access-name. Otherwise, entity syntax shall be displayed as specified in the sort-clause.
  - (3) The show-relationships-syntax-clause shall be used to determine whether relationships shall be displayed after each entity, or whether all relationships shall be displayed after all entities are displayed:
    - (a) If this command-clause is not specified, or if:

SHOW RELATIONSHIP SYNTAX FOR EACH [ ENTITY ]

is specified, then all relationships in which the selected entity participates shall be presented after the selected entity's syntax and before the next entity's syntax shall be shown.

For each selected entity, relationships shall be presented in the following order:

relationship-type,  
 first-entity-name,  
 sequence-attribute or ORDER (if applicable),  
 second-entity-name.

- (b) If show-relationships-syntax-clause specifies:

SHOW RELATIONSHIP SYNTAX AFTER [ LAST ENTITY ]

then all relationships shall be presented after the last entity. Here relationships shall be presented in the following order:

relationship-type,  
 first-entity-name,  
 sequence-attribute or ORDER (if applicable),  
 second-entity-name.

In the former case, syntax for relationships may be duplicated. In the latter case, no relationships shall be duplicated.

- (4) If show-IRD-life-cycle-phase-clause is specified, then for each entity selected, the IRD Life-Cycle-Phase which contains it shall be displayed within a comment. If this command-clause is not specified, the IRD Life-Cycle-Phase shall not be displayed.
- (5) If no show-relationships-clause is specified, every relationship which contains a selected entity shall be presented, provided it is of a type which is visible via the IRD-Views used in the command.
- (6) The options available in this command via show-relationships-clause shall be restricted to the following:
  - (a) Show all relationships.
  - (b) Show no relationships.
  - (c) Show relationships of specified relationship-types.
- (7) The implementor shall provide the syntax in a readable format. It is suggested that blank lines be used to separate individual entities and/or relationships. It is also suggested that attribute-type-designators and attribute-group-type-designators be indented.
- (8) The implementor may, at his option, also use comments to indicate control breaks, enhance readability, and provide additional information (such as audit-attributes and control breaks by entity-types or relationship-types). Implementor defined command-clauses may be used to provide user control of the degree of such commenting.

**Security Rules** None.

### **Actions**

- (1) As specified in Actions of "IRD Output Commands", Subsection 5.2.2.

### **Error and Warning Conditions**

- (1) Error E01108: Duplicated command-clause. Syntax Rule (1) has been violated.
- (2) Error E01110: Command-clause option invalid for command. See General Rule (6).

NOTE: The output-syntax-command shall be necessary and meaningful only in the event that the Command Language Interface is available.

## **5.3 General Commands**

**Function** To identify commands which are neither IRD Schema commands nor IRD commands.

### **Format**

/\* None. This breakdown is explanatory only. \*/

```
general-command ::=
    ird-ird-interface-commands      5.3.1
    | utility-commands              5.3.2
```



**Syntax Rules** None.

**General Rules** None.

**Security Rules** None.

**Actions** None.

**Error and Warning Conditions** None.

### 5.3.1 IRD-IRD Interface Commands

**Function** These commands provide for extracting the IRD Schema and contents of one IRD and incorporating them into another IRD.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

ird-ird-interface-command ::=
    create-IRD-command           5.3.1.1
    | export-IRD-command         5.3.1.2
    | check-IRD-schema-compatibility-command 5.3.1.3
    | import-IRD-command        5.3.1.4

```

**Syntax Rules** None.

**General Rules** None.

**Security Rules** None.

**Actions** None.

**Error and Warning Conditions** None.

#### 5.3.1.1 Create IRD Command

**Function** This command shall be used to create a new IRD of a specified name with an initial IRD Schema, and optionally load it with the contents of a IRD-export-file.

**Format**

```

create-IRD-command ::=
    create-word           4.3
    IRD-word             4.3
    new-IRD-name
    [ location-clause ]  6.57
    IRD-schema-source-clause 6.53
    [ load-IRD-clause ] **EX/IM** 6.54
    [ implementor-defined-clauses ]
    ;

```

```

new-IRD-name ::=
    /* implementor-defined-format */

```

```
implementor-defined-clauses ::=  
/* Consult General Rule (2) */
```

### Syntax Rules

- (1) The new-IRD-name shall conform to the implementor-defined format for IRD names.
- (2) Load-IRD-clause shall be specified only if IRD-schema-source-clause specifies an IRD-schema-export-file. **\*\*EX/IM\*\***

### General Rules

- (1) New-IRD-name shall not specify the name of an existing IRD.
- (2) Additional implementor-defined command-clauses may be provided with this command. These command-clauses may be used to define any implementation-dependent options, such as device-dependent characteristics. The implementor shall identify whether each such command-clause is mandatory or optional.
- (3) The IRD-schema-export-file identified in source-IRD-schema-clause and the IRD-export-file identified in load-IRD-clause shall have been created by the same export-IRD-command. This shall be verified by matching the file-titles of each file. In particular, the file titles of each of the two files shall specify the same IRD-name, date, time, and suffix (if present). Consult the export-IRD-command (Subsection 5.3.1.2) for details on the contents of each of these files.

**Security Rules**    None.

### Actions

- (1) An IRD Schema shall be created. The newly created IRD Schema shall be the same as the IRD Schema specified by IRD-schema-location except for audit-meta-attributes and IRD-PARTITIONS. The audit-meta-attribute-group DATE-TIME-ADDED and the audit-meta-attribute ADDED-BY shall be inserted with appropriate values. Four IRD-PARTITIONS shall also be created in the IRD Schema.
  - (a) If load-IRD-clause is not specified, there shall be one IRD-PARTITION with an assigned-access-name of SECURITY. The remaining IRD-PARTITIONS define IRD life-cycle-phases. One IRD life-cycle-phase has a LIFE-CYCLE-PHASE-CLASS of UNCONTROLLED, one has a LIFE-CYCLE-PHASE-CLASS of CONTROLLED, and one has a LIFE-CYCLE-PHASE-CLASS of ARCHIVED. The names assigned to these life-cycle-phases shall be implementor-defined.
  - (b) If load-IRD-clause is specified, the IRD-PARTITIONS identified in the corresponding IRD-schema-export-file shall be established in the IRD Schema of the newly created IRD.
- (2) An IRD with the above IRD Schema shall be created. One master IRDS-USER entity shall be created, with an implementor-assigned access-name. Additionally, one IRD-VIEW for each IRD-PARTITION shall be created and related to the master IRDS-USER entity. The IRDS-USER and IRD-VIEW entities shall reside in the IRD-PARTITION named SECURITY. The default IRD-PARTITION for the master IRDS-USER entity shall be assigned according to an implementation-dependent rule.

If no security descriptors shall be loaded with the create-IRD-command, then the following entities and attributes shall be loaded into the IRD:

- (a) An entity of type IRDS-USER with assigned-access-name ADMINISTRATOR.
- (b) An entity of type IRD-VIEW with assigned-access-name ADMINISTRATOR-IRD-VIEW, with the attribute of type IRD-PARTITION-NAME equal to SECURITY.
- (c) An entity of type IRD-SCHEMA-VIEW with assigned-access-name ADMINISTRATOR-IRD-SCHEMA-VIEW, with the attribute of type IRD-SCHEMA-PHASE-NAME equal to CONTROLLED.

Furthermore, the following relationships and associated attributes shall also exist in the IRD:

- (d) A relationship of type IRDS-USER-HAS-IRD-VIEW with members ADMINISTRATOR and ADMINISTRATOR-IRD-VIEW, with the attribute of type DEFAULT-VIEW equal to YES.
  - (e) A relationship of type IRDS-USER-HAS-IRD-SCHEMA-VIEW with members ADMINISTRATOR and ADMINISTRATOR-IRD-SCHEMA-VIEW, with the attribute of type DEFAULT-VIEW equal to YES.
- (3) If load-IRD-clause is specified, then the newly created IRD shall be loaded with the contents of the specified IRD-export-file. Each entity within the IRD-export-file shall be added to the IRD-PARTITION in which it resides in the source IRD. Revision-numbers associated with entity access-names shall be preserved. However, all audit-attributes shall be reset. The audit-attribute-group-type DATE-TIME-ADDED and the audit-attribute-type ADDED-BY shall identify the date and time of the load and the IRDS-USER who performed the load, respectively. The NUMBER-OF-TIMES-MODIFIED attributes shall all be reset to zero, and the DATE-TIME-LAST-MODIFIED attribute-group and LAST-MODIFIED-BY attribute shall be null.

#### **Error and Warning Conditions**

- (1) Error E01039: Invalid IRD name. Syntax Rule (1) for this command has been violated.
- (2) Error E01040: Inconsistent command-clauses. Syntax Rule (2) for this command has been violated.
- (3) Error E01041: New-IRD-name already exists. General Rule (1) for this command has been violated.
- (4) Error E01043: IRD-schema-export-file is not synchronized with IRD-export-file. General Rule (3) for this command has been violated.

#### **5.3.1.2 Export IRD Command \*\*EX/IM\*\***

**Function** This command shall be used to obtain the IRD Schema and a subset of the IRD contents from an IRD and place them in specified data files in IRD Schema export format and IRD export format, respectively.

**Format**

```

export-IRD-command ::=
    export-word           4.3
    IRD-word             4.3
    [ using-IRD-views-clause ] 6.38
    extraction-option
    [ exclude-relationship-of-type-clause ] 6.60
    IRD-schema-export-file-clause 6.55
    [ controlled-only-option ]
    [ IRD-export-file-clause ] 6.56
    [ file-title-suffix-clause ] 6.58
    [ implementor-defined-clauses ]
    ;

extraction-option ::=
    using-list-clause    6.1
    | IRD-schema-only-option

IRD-schema-only-option ::=
    IRD-schema-word     4.3
    only-word           4.3

controlled-only-option ::=
    controlled-word     4.3
    only-word           4.3

implementor-defined-clauses ::=
    /* Consult General Rule (5) */

```

**Syntax Rules**

- (1) If extraction-option does not specify IRD-schema-only-option, IRD-export-file-clause shall be specified.

**General Rules**

- (1) If extraction-option specifies IRD-schema-only-option, the IRD-export-file-clause shall be ignored if specified.
- (2) All relationships for which both members exist in the specified entity-list shall be extracted, except for those of the type or class-type specified in the exclude-relationship-of-type-clause.
- (3) The IRD-schema-export-file specified in the IRD-schema-export-file-clause shall not be the name of an existing file.
- (4) If controlled-only-option is specified, then only those IRD Schema descriptors in the CONTROLLED IRD-schema-life-cycle-phase shall be extracted. Otherwise, all IRD Schema descriptors shall be extracted, regardless of IRD-schema-life-cycle-phase.
- (5) The IRD-export-file specified in the IRD-export-file-clause shall not be the name of an existing file.

- (6) Additional implementor-defined command-clauses may be provided with this command. These command-clauses may be used to define device-dependent characteristics, implementation-dependent options, or for any other appropriate use. The implementor shall identify whether each such command-clause is mandatory or optional.
- (7) If using-IRD-views-clause is not directly specified in the command, the IRD-View in effect shall be used by the command.
- (8) The IRD being imported to shall be in the "activated" state (i.e., shall not have been deactivated via a deactivate-IRD-command).

### Security Rules

- (1) The IRD-Views used by this command:
  - (a) Shall be used for limiting the scope of selection within using-new-list-clause. A warning condition shall be raised if the selection-criteria within using-new-list-clause explicitly identify entities not within the IRD-Views being used by the command.
  - (b) Shall restrict the exporting of entities identified within an entity-list to only those entities visible within the specified IRD-Views. Entities identified within the entity-list which are not visible within the IRD-Views used in the command shall be bypassed. In this event a warning condition shall be raised.

### Actions

- (1) Two files, the IRD-schema-export-file and the IRD-export-file shall be produced, unless IRD-schema-only-option is specified. In that event, only the IRD-schema-export-file shall be produced. The formats of both the IRD-schema-export-file and the IRD-export-file, shall be in the format specified in the IRDS Export/Import Format Standard. The character representation used shall be standard ASCII.
- (2) The IRD-schema-export-file specified in the IRD-schema-export-file-clause shall be loaded either with all IRD Schema descriptors or only those in the CONTROLLED IRD-schema-life-cycle-phase (depending on whether the controlled-only-option was specified) in the IRD in which the user is operating.
- (3) The IRD-export-file specified in the IRD-export-file-clause will be loaded with the IRD entities specified in the extraction-option and with IRD relationships as specified in General Rule (2).

### Error and Warning Conditions

- (1) Error E01044: Invalid, missing, or misplaced command-clause. Syntax Rule (1) of this command has been violated.
- (2) Error E01045: The IRD-schema-export-file-name specified in the IRD-schema-export-file-clause already exists. General Rule (3) has been violated.
- (3) Error E01046: The IRD-export-file-name specified in the IRD-export-file-clause already exists. General Rule (5) has been violated.
- (4) Warning W01009: Specified entities bypassed. See Security Rule (1).

**5.3.1.3 Check IRD Schema Compatibility Command \*\*EX/IM\*\***

**Function** To determine if the IRD Schema contained either in another IRD or an IRD export format file is compatible with the IRD Schema of the IRD which the user is accessing. If the identified IRD Schema is compatible with the IRD Schema of the IRD in use, IRD descriptors can be transferred from one IRD to another.

**Format**

```

check-IRD-schema-compatibility-command ::=
    check-word                4.3
    IRD-schema-word          4.3
    other-IRD-schema-clause  6.59
    [ implementor-defined-clauses ]
    ;

implementor-defined-clauses ::=
    /* Consult General Rule (5) */
    
```

**Syntax Rules** None.

**General Rules**

- (1) One of the IRD Schemas involved in the check-IRD-schema-compatibility-command shall be the IRD Schema of the IRD which is in use.
- (2) Other-IRD-schema-clause identifies:
  - (a) Whether the other IRD Schema is the MINIMAL IRD Schema or if it is located in an IRD-schema-export-file or another IRD, and
  - (b) Whether the other IRD Schema is to act as the source-IRD Schema or target-IRD Schema. The source-IRD Schema is the IRD Schema of the IRD from which the IRD-schema-export-file and IRD-export-file shall be created. The target-IRD Schema is the IRD Schema of the IRD which describes the IRD descriptors contained in the IRD-export-file.
- (3) Designating the other IRD Schema as a source-IRD Schema implies that the IRD Schema of the IRD in use shall be regarded as the target-IRD Schema. Conversely, designating the other IRD Schema as a target-IRD Schema implies that the IRD Schema of the IRD in use is the source-IRD Schema.
- (4) The source-IRD Schema is said to be compatible with the target-IRD Schema if and only if:
  - (a) Sets of meta-entities shall be compared, for which each of the following conditions apply:
    - (i) Each meta-entity belongs to one of the following types:
      - ENTITY-TYPE
      - RELATIONSHIP-CLASS-TYPE
      - RELATIONSHIP-TYPE
      - ATTRIBUTE-GROUP-TYPE

ATTRIBUTE-TYPE  
 ATTRIBUTE-TYPE-VALIDATION-DATA  
 ATTRIBUTE-TYPE-VALIDATION-PROCEDURE  
 VARIATION-NAMES-DATA  
 QUALITY-INDICATOR  
 IRDS-DEFAULTS

- (ii) Each meta-entity shall be in the CONTROLLED IRD Schema life-cycle-phase.

The set of meta-entities as defined above is called the IRD Schema comparison set. The sets for the source and target IRD Schemas are called the source-IRD Schema comparison set and target-IRD Schema comparison set respectively. Two meta-entities, one in each of the above sets, are said to be comparable if they have the same assigned-access-name. Each meta-entity in the source-IRD Schema comparison set shall have a comparable meta-entity in the target-IRD Schema comparison set.

- (b) Each meta-relationship which exists in the source-IRD Schema where both meta-entities have a meta-entity-type listed above also exists in the target-IRD Schema.
- (c) The following conditions apply to RELATIONSHIP-TYPE meta-entities:
- (i) The value of the meta-attribute SEQUENCED shall be the same for each RELATIONSHIP-TYPE meta-entity in each IRD Schema.
- (ii) For each such RELATIONSHIP-TYPE where the meta-attribute SEQUENCED is YES, the value of the SEQUENCE-PARAMETER meta-attribute shall be the same in each IRD Schema. The SEQUENCE-PARAMETER meta-attribute-type is defined on the RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship-type.
- (d) The following conditions apply to ATTRIBUTE-TYPE meta-entities:
- (i) The MINIMUM-ATTRIBUTE-LENGTH meta-attribute in the source-IRD Schema shall be greater than or equal to the corresponding meta-attribute in the target-IRD Schema.
- (ii) The MAXIMUM-ATTRIBUTE-LENGTH meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
- (e) The following conditions apply to ATTRIBUTE-TYPE-VALIDATION-DATA meta-entities:
- (i) The VALIDATION-TYPE meta-attributes shall be the same.
- (ii) Every DATA-VALUE meta-attribute-group of an ATTRIBUTE-TYPE-VALIDATION-DATA meta-entity in the source-IRD Schema shall have a corresponding equal DATA-VALUE meta-attribute-group for the corresponding meta-entity in the target-IRD Schema.
- (iii) Every DATA-RANGE meta-attribute-group of an ATTRIBUTE-TYPE-VALIDATION-DATA meta-entity in the source-IRD Schema shall be a valid subrange of a DATA-RANGE meta-attribute-group of the corresponding ATTRIBUTE-

TYPE-VALIDATION-DATA in the target-IRD Schema. To be a valid sub-range, the LOW-VALUE meta-attribute in the source-IRD Schema shall be greater than or equal to the LOW-VALUE meta-attribute in the target-IRD Schema and for the same pair of DATA-RANGE meta-attribute-groups, the HIGH-VALUE meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding HIGH-VALUE meta-attribute in the target-IRD Schema.

- (f) For corresponding VARIATION-NAMES-DATA meta-entities, each VARIATION meta-attribute in the source-IRD Schema shall also exist in the target-IRD Schema.
- (g) For corresponding ENTITY-TYPE meta-entities:
  - (i) The MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH meta-attribute in the source-IRD Schema shall be greater than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (ii) The MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (iii) The MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH meta-attribute in the source-IRD Schema shall be greater than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (iv) The MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
- (h) For corresponding IRDS-LIMITS meta-entities:
  - (i) The MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LIMIT meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (ii) The MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LIMIT meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (iii) The MAXIMUM-META-ENTITY-ASSIGNED-ACCESS-NAME-LIMIT meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (iv) The MAXIMUM-META-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LIMIT meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (v) The MAXIMUM-NUMBER-OF-OCCURRENCES-LIMIT meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (vi) The INTEGER-LIMIT meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.



- (vii) The **LINE-COUNT-LIMIT** meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (viii) The **LINE-LENGTH-LIMIT** meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (ix) The **STRING-LENGTH-LIMIT** meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (x) The **VARIATION-NAME-LENGTH-LIMIT** meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
- (i) For corresponding IRDS-DEFAULTS meta-entities:
- (i) The **MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-DEFAULT** meta-attribute in the source-IRD Schema shall be greater than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (ii) The **MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-DEFAULT** meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (iii) The **MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-DEFAULT** meta-attribute in the source-IRD Schema shall be greater than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (iv) The **MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-DEFAULT** meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
  - (v) The **MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT** meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target-IRD Schema.
- (j) For corresponding meta-relationships of meta-relationship-type (**RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE**), the **POSITION** meta-attributes shall be the same.
- (k) For corresponding meta-relationships of meta-relationship-type (**ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE**) the **GROUP-POSITION** meta-attributes shall be the same.
- (l) For corresponding meta-relationships of meta-relationship-type (**RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE**) or (**RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE**) or (**ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE**) or (**ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE**):
- (i) If the **SINGULAR** meta-attribute is **NO** in the source-IRD Schema it shall be **NO** in the target-IRD Schema.

- (ii) The MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute in the source-IRD Schema shall be less than or equal to the corresponding meta-attribute in the target IRD Schema.
- (5) The implementor may define any additional command-clauses as required. The implementor shall have the right to define all such command-clauses as mandatory or optional. Such command-clauses may be used for purposes such as allocating work space or establishing authorization for the user to read an IRD Schema of another IRD.
- (6) A warning message shall be provided if both IRDS's whose IRD Schemas are being checked do not have a value of YES for the meta-attribute-type STANDARD-MODE.

### Security Rules

- (1) If in-IRD-clause is specified in the other-IRD-schema-clause, the user shall establish authorization to read the IRD Schema. The mechanism for establishing such authorization is implementor-defined. Implementor-defined-clauses may be used for this purpose.

### Actions

- (1) If the command passes validation, the source-IRD Schema is checked for compatibility with the target-IRD Schema. If the two IRD Schemas are compatible, a message shall be issued confirming this. If any incompatibilities are detected, all appropriate messages shall be issued. The implementation shall clearly identify the source of each incompatibility. If incompatibilities are encountered, the system shall also clearly indicate completion of processing.

### Error and Warning Conditions

- (1) Error E01048: Access to other IRD Schema denied. See Security Rule (1) of this section.
- (2) Warning W01002: One or both of the IRDS's are not in STANDARD mode. See General Rule (6).

The following conditions are classified as errors because they may occur as part of the IRD Schema compatibility check performed in the import-IRD-command. They are not error conditions of the check-IRD-schema-compatibility-command, but diagnostics.

- (3) Error E01049: Meta-entity exists in source-IRD Schema but not in target-IRD Schema. See General Rule (4)(a).
- (4) Error E01050: Meta-relationship exists in source-IRD Schema but not in target-IRD Schema. See General Rule (4)(b).
- (5) Error E01051: An attribute-type or attribute-group-type is NO in the source-IRD Schema but YES in target-IRD Schema. See General Rule (4)(l)(i).
- (6) Error E01052: Source-IRD Schema DATA-RANGE is not a subrange of corresponding target-IRD Schema DATA-RANGE. See General Rule (4)(e)(iii).
- (7) Error E01053: Source-IRD Schema meta-attribute does not exist in target-IRD Schema. See various conditions.

- (8) Error E01061: Source-IRD Schema and target-IRD Schema meta-attributes not identical. See various conditions.
- (9) Error E01054: Source-IRD Schema meta-attribute must be greater than or equal to corresponding target-IRD Schema meta-attribute. See various conditions.
- (10) Error E01055: Source-IRD Schema meta-attribute must be less than or equal to corresponding target-IRD Schema meta-attribute. See various conditions.

#### 5.3.1.4 Import IRD Command **\*\*EX/IM\*\***

**Function** This command is used to import the contents of an IRD-export-file into an IRD.

#### **Format**

```

import-IRD-command ::=
    import-word                4.3
    IRD-word                   4.3
    IRD-schema-export-file-clause 6.55
    IRD-export-file-clause     6.56
    IRD-life-cycle-phase-designation-clause 6.61
    [ implementor-defined-clauses ]
    ;

implementor-defined-clauses ::=
    /* Consult General Rule (6) */

```

**Syntax Rules** None.

#### **General Rules**

- (1) IRD Schema compatibility between the IRD in which the user is operating and the specified IRD-schema-export-file will be checked according to the rules defined in the check-IRD-schema-compatibility-command.
- (2) The IRD-schema-export-file specified in the IRD-schema-export-file-clause shall exist at the specified or assumed location.
- (3) The IRD-export-file specified in the IRD-export-file-clause shall exist at the specified or assumed location.
- (4) The IRD life-cycle-phase specified in the IRD-life-cycle-phase-designation-clause shall designate an IRD-PARTITION, other than the IRD-PARTITION named SECURITY, which represents an empty, UNCONTROLLED life-cycle-phase.
- (5) Additional implementor-defined command-clauses may be provided with this command. These command-clauses may be used to define device-dependent characteristics, implementation-dependent options, or for any other appropriate use. The implementor shall identify whether each such command-clause is mandatory or optional.
- (6) The identified IRD-schema-export-file and IRD-export-file shall have been created in the same command. In particular the IRD-name, date, time and optional string (if present) shall match.

- (7) The IRD being imported to shall be in the "activated" state (i.e., shall not have been deactivated via a deactivate-IRD-command).

### Security Rules

- (1) Authority to issue the import-IRD-command requires an IRDS-USER entity which is related to an IRD-SCHEMA-VIEW entity.

### Actions

- (1) The specified IRD-schema-export-file will be checked for compatibility against the IRD in which the user is operating.
- (2) The contents of the specified IRD-export-file will be imported into the IRD life-cycle-phase specified in the IRD-life-cycle-phase-designation-clause.
- (3) The revision-number of each entity being imported will be checked against existing entities with the same assigned-access-name and variation-name in the IRD in which the user is operating. If revision-number conflicts are found, the revision-number of the entity being imported will be adjusted to one greater than the highest current revision-number associated with any entity having the same assigned-access-name and variation-name.
- (4) The assigned-access-name and variation-name of each entity being imported are checked for potential conflict with system-generated entity access-names in the IRD in which the user is operating. The method for checking shall be the same as that discussed for the add-entity-command. If a potential conflict exists, the entity shall be written to an error file.

### Error and Warning Conditions

- (1) Error E01056: Incompatible IRD Schemas. General Rule (1) has been violated.
- (2) Error E01057: The IRD-schema-export-file-name specified in the IRD-schema-export-file-clause does not exist. General Rule (2) has been violated.
- (3) Error E01058: The IRD-export-file specified in the IRD-export-file-clause does not exist. General Rule (3) has been violated.
- (4) Error E01060: Cannot import into specified IRD life-cycle-phase. General Rule (4) and/or General Rule (5) has been violated.
- (5) Error E01042: Access to IRD Schema denied. Security Rule (1) has been violated.
- (6) Error E01043: IRD-schema-export-file is not synchronized with IRD-export-file. General Rule (7) for this command has been violated.

### 5.3.2 Utility Commands

**Function** These commands are used to manage a session.

#### Format

/\* None. This breakdown is explanatory only. \*/

```

utility-command ::=
    set-session-defaults-command          5.3.2.1
    | session-status-command              5.3.2.2
    | help-command                        5.3.2.3
    | exit-IRDS-system-command            5.3.2.4
    | enter-panel-dialogue-command        5.3.2.5

```

**Syntax Rules**    None.

**General Rules**    None.

- (1) With the exception of the exit-IRDS-system-command, each of the utility-commands require that the IRD be in the "activated" state (i.e., shall not have been deactivated via a deactivate-IRD-command).

**Security Rules**    None.

**Actions**        None.

**Error and Warning Conditions**    None.

### 5.3.2.1    Set Session Defaults Command

**Function**        To set or reset defaults for or within a session.

**Format**

```

set-session-defaults-command ::=
    set-word          4.3
    options-list
    [ save-word ]    4.3
    ;

options-list ::=
    option
    [ options-list ]

option ::=
    effective-view-option
    | attribute-decoding-option
    | implementor-defined-option

effective-view-option ::=
    IRD-word | IRD-schema-word  4.3
    view-word                    4.3
    =
    security-entity-name        4.8

attribute-decoding-option ::=
    show-word          4.3
    attributes-word    4.3
    encoded-or-decoded

```

```
encoded-or-decoded ::=
    encoded-word           4.3
  | decoded-word          4.3

implementor-defined-option ::=
    /* implementor-defined-format */
```

### Syntax Rules

- (1) No option shall be specified more than once within a command.

### General Rules

- (1) The effective-view-option shall be used to establish an IRD-View or an IRD-Schema-View as the effective IRD-View or the effective IRD-Schema-View for the session. The specified view shall remain the effective view until either the end of the session or another set-session-defaults command shall be issued.
- (2) The IRD-View or IRD-Schema-View specified in the effective-view-option shall be related to the effective IRDS-user.
- (3) Attribute-decoding-option defines whether attributes shall be displayed in their encoded (i.e., input) form or decoded form in outputs. This option shall have no effect on the output-syntax-command.
- (4) The implementor may provide other options (e.g., default routing).
- (5) Save-word indicates that the changed defaults shall be made permanent and the appropriate changes be made in the IRD. In this case, the specified defaults shall be used in subsequent sessions unless specifically overridden. Otherwise, the defaults shall be changed as specified only for the duration of the session or the next set-session-defaults command within the current session.

**Security Rules**    None.

### Actions

- (1) The command shall be validated. If any errors are encountered, the appropriate messages shall be given.
- (2) If the command passes validation, the session defaults shall be modified as specified. If save-word is specified, the session defaults shall be saved.
- (3) Confirmation of command completion shall be given.

### Error and Warning Conditions

- (1) Error E01044: Invalid, missing, or misplaced command-clause. An option incorrectly specified.
- (2) Error E01108: Duplicated command-clause. Syntax Rule (1) has been violated.
- (3) Error E01111: Invalid IRD-View. General Rule (2) has been violated.

- (4) Error E01112: Invalid IRD-Schema-View. General Rule (2) has been violated.

### 5.3.2.2 Session Status Command

**Function** To display session-related information.

#### Format

```

session-status-command ::=
    status-word                4.3
    all-or-status-options

all-or-status-options ::=
    all-word                    4.3
    | status-options-list

status-option-list ::=
    status-option
    [ , status-option-list ]

status-option ::=
    IRD-word                    4.3
    | entity-list-word          4.3
    | views-option              4.3
    | defaults-word             4.3
    | implementor-defined-option

views-option ::=
    IRD-word | IRD-schema-word  4.3
    views-word                    4.3

implementor-defined-option ::=
    /* implementor-defined-format */

```

#### Syntax Rules

- (1) No status option shall be specified more than once.

#### General Rules

- (1) This command shall not be executable in batch mode. If specified in batch mode, it shall be ignored.
- (2) The meanings of the various options are as follows:
- (a) IRD-word under status-option shall cause the name of the IRD against which the user is operating to be displayed.
  - (b) Entity-list word shall cause all entity-lists which currently exist with the session to be displayed. This display shall be the same as specified in the output-entity-list-names command.

- (c) Views-option shall cause the names of all IRD-View or IRD-Schema-View entities which are associated with the effective IRDS-USER to be displayed. Within this display, the effective IRD-View or IRD-Schema-View shall be clearly identified.
- (d) Defaults-word shall cause all defaults which can be set via the set-session-options command to be displayed.

- (3) The implementor may define additional options.
- (4) All-word shall be equivalent to specifying all of the options individually.

**Security Rules** None.

**Actions**

- (1) If any errors are encountered, all appropriate messages shall be issued. If specified in a batch command stream, the command shall be ignored.
- (2) The requested session-related information shall be displayed.
- (3) Command completion shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01121: Invalid or unrecognized option. An option incorrectly specified.
- (2) Error E01122: Duplicate option. Syntax Rule (1) has been violated.

**5.3.2.3 Help Command**

**Function** To obtain help.

**Format**

```
help-command ::=
    help-word                                     4.3
    [ help-option ]
    ;

help-option ::=
    message-option
    | all-word                                     4.3
    | command-imperative-substring-option

message-option ::=
    message-word                                   4.3
    [ message-identifier ]

command-imperative-substring-option ::=
    command-imperative-word                       4.3
    [ command-imperative-substring ]
```



```

command-imperative-substring ::=
    /* A command-imperative-substring is a sequence of irds-words
       which are used in IRDS command-imperatives */

message-identifier ::= /* implementor defined */

```

### Syntax Rules

- (1) If help-option is not specified, then all-word shall be assumed.

### General Rules

- (1) If all-word is specified or assumed, then the help command shall display the inventory of command-imperatives, using the full form of the irds-words.
- (2) Message-option shall cause an explanation of an error or warning message to be displayed.
  - (a) If only message-word is specified, then an explanation of all error messages encountered in the prior command shall be displayed. It is suggested that the implementation prompt the user for an explanation of each message issued when multiple errors have occurred. If such prompting is provided, the implementation shall permit the user to bypass all messages or select one of the messages issued.
  - (b) If a message-identifier is specified after message-word, then an explanation of the identified message shall be displayed.
  - (c) If an invalid message-identifier is specified after message-word, then all message-identifiers encountered in the prior command shall be displayed. It is suggested that the implementation prompt the user as in (2)(a) above.
- (3) If command-imperative-substring is specified, the system shall attempt to match the specified words against the words used in the command-imperatives.
  - (a) If the specified words match a command-imperative, then an explanation of the command shall be given.
  - (b) If the specified words match irds-words within several commands, then the command-imperatives of each of those commands shall be displayed.
  - (c) If the sequence of words does not partially or completely match the irds-words within one or more command-imperatives, a full list of all command-imperatives shall be displayed, along with instructions on selecting a command.
- (4) If help-option is incorrectly specified, the explanation of the help-command shall be displayed, with instructions to try again.
- (5) This command shall be ignored in batch mode.

**Security Rules**    None.

### Actions

- (1) Help shall be provided in accordance with the above rules.

**Error and Warning Conditions** None.

### 5.3.2.4 Exit IRDS Command

**Function** To exit the control of the IRDS.

**Format**

```
exit-IRDS-system-command ::=
    exit-word
    ;
```

4.3

**Syntax Rules** None.

**General Rules**

- (1) This command shall be ignored in batch mode.

**Security Rules** None.

**Actions**

- (1) Control shall be returned to the system which invoked the IRDS.

**Error and Warning Conditions** None.

### 5.3.2.5 Enter Panel Dialogue Command

**Function** To enter the IRDS Panel Interface at a specified point.

**Format**

```
enter-panel-dialogue-command ::=
    panel-word
    [ panel-transfer-option ]
    ;
```

4.3

```
panel-transfer-option ::=
    panel-name-option
    | command-imperative
```

4.3

```
panel-name-option ::=
    name-word
    =
    panel-name
```

4.3

```
panel-name ::=
    /* implementor-defined-format */
```

**Syntax Rules** None.

**General Rules**

- (1) If no panel-transfer-option is specified, then control shall be transferred to the "HOME" panel.
- (2) The panel-name shall be a valid panel-name. If panel-name is incorrect, a warning condition shall be raised. Control shall be transferred to the "HOME" panel.
- (3) The command-imperative specified shall be for a command which is supported by the Panel Interface. If the command-imperative does not meet this condition, an error condition exists.
- (4) This command shall be ignored in batch mode.

**Security Rules**     None.

**Actions**

- (1) The command shall be validated. If an error is encountered, the system shall issue the error, and provide the user with help for the panel command.
- (2) If the warning condition of General Rule (2) exists, control shall be transferred to the "HOME" panel. The warning shall be shown in the message-area.
- (3) If neither of the above conditions apply, control shall be transferred to the Panel Interface. The specified or assumed panel shall be presented to the user.

**Error and Warning Conditions**

- (1) Warning W01019: Incorrect panel-name specified. Control transferred to "HOME" panel. See General Rule (2).
- (2) Error E01123: Incorrect command-imperative. See General Rule (3).
- (3) Error E01124: Command is not supported by Panel Interface. See General Rule (3).

**6 Command-Clause Specifications**

The order of command-clause presentation is not significant. An alphabetical index of command-clauses is provided (Subsection 7.2) to assist referencing any command-clause. Note that gaps exist in the numbering of command-clauses.

**6.1 Entity-Type Qualification Command-Clause**

**Function**     To identify the entity-type(s) which will qualify subsequent restriction-clauses and show-clauses.

**Format**

```

entity-type-qualification-clause ::=
    for-word                               4.3
    [ entities-word                         4.3
      of-word ]                             4.3
    entity-type-list

entity-type-list ::=
    entity-type-designator                 4.6
    [ , entity-type-list ]

```

**Syntax Rules**

- (1) Each entity-type-designator shall identify an ENTITY-TYPE meta-entity in the IRD Schema.

**General Rules**

- (1) For-word denotes qualification. A qualification-clause shall be either an entity-type qualification-clause or a relationship-type qualification-clause. The attribute-group-restriction-clause also begins with for-word. Note that an entity-type qualification shall be the highest level of qualification, and other qualification-clauses shall be governed by an entity-type qualification-clause. (See Subsection 6.33, "Entity Selection Criteria-clause", for a discussion of "governing".)
- (2) If entities-word and of-word are not specified after for-word, then the command-clause shall be determined by the first designator after for-word.

**Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. An IRD-schema-descriptor was expected. Designator is not a meta-entity in the IRD Schema.
- (2) Error E01002: Invalid or unrecognized entity-type. Entities-word and of-word followed for-word, or the first designator in the list identified a valid ENTITY-TYPE. Designator in list does not identify an ENTITY-TYPE meta-entity in the IRD Schema.
- (3) Error E01127: Improperly qualified command-clause. IRD-schema-descriptor was not used for qualification. Note that any designator which follows for-word shall either be an irds-word, or an entity-type-designator, relationship-type-designator, or attribute-group-type-designator.

NOTE: This command-clause corresponds to the meta-entity-type-qualification IRD-schema-clause in Subsection 6.120.

**6.2 Entity-Type Command-Clause**

**Function** To identify the entity-type for an entity in a add-entity-command, and to document the entity-type of an existing entity in a modify-entity-command.

**Format**

entity-type-clause ::=	
entity-type-word	4.3
=	
entity-type-designator	4.6

**Syntax Rules**

- (1) Entity-type-designator shall identify a valid ENTITY-TYPE in the IRD Schema.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01002: Invalid or unrecognized entity-type. Syntax Rule (1) for this command-clause has been violated.

NOTE: This command-clause corresponds to the meta-entity-type IRD-schema-clause in Subsection 6.101.

**6.3 Simple Attribute Command-Clause**

**Function** To declare for an entity or a relationship:

- (1) A new nonrepeating attribute, or
- (2) A new value for an existing nonrepeating attribute, or
- (3) To erase an existing nonrepeating attribute.

**Format**

simple-attribute-clause ::=	
attribute-type-designator	4.6
=	
non-text-attribute	4.5

**Syntax Rules**

- (1) The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema. Its FORMAT meta-attribute shall not have a value of TEXT.
- (2) The ATTRIBUTE-TYPE identified by attribute-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being created or modified by the command.
- (3) If the command operates on an entity, the SINGULAR meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship in the IRD Schema shall have a value of YES. If the command operates on a relationship, the SINGULAR meta-attribute of the corresponding RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship in the IRD Schema shall have a value of YES.

- (4) The attribute shall conform to all validation rules defined in the IRD Schema for the corresponding ATTRIBUTE-TYPE.
- (5) Attribute-type-designator shall not be the name of an audit attribute-type.

### **General Rules**

- (1) This command-clause shall be specified only once for each ATTRIBUTE-TYPE within a command.
- (2) If the attribute is null-mark, then if the attribute exists it will be erased. If the attribute does not exist, no action will be taken and a warning message shall be issued.

### **Error and Warning Conditions**

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01148: Attribute not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01128: Invalid attribute or format. Syntax Rule (4) of this command-clause has been violated.
- (4) Warning W01020: Attribute does not exist and cannot be erased. A null-mark was specified as an attribute for an attribute which does not exist.
- (5) Error E01129: Attribute specified more than once in command. General Rule (1) of this command-clause has been violated.
- (6) Error E01130: Declaration of value for audit attribute not allowed. Syntax Rule (5) of this command-clause has been violated.
- (7) Error E01131: Repeating attribute specified. Syntax Rule (3) of this command-clause has been violated.

### **NOTES:**

- (1) If Syntax Rule (4) of this command-clause is violated, the error message(s) shall clearly identify which validation rules were violated.
- (2) This command-clause corresponds to the simple-meta-attribute IRD-schema-clause in Subsection 6.106.

## **6.4 New Repeating Attribute Command-Clause**

**Function** To declare a repeating attribute for an entity or a relationship when the entity or relationship is created.

**Format**

new-repeating-attribute-clause ::=	
attribute-type-designator	4.6
=	
repeating-attribute-list	4.5

**Syntax Rules**

- (1) The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema.
- (2) The ATTRIBUTE-TYPE identified by attribute-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being created by the command.
- (3) If the command operates on an entity, the SINGULAR meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship in the IRD Schema shall have a value of NO. If the command operates on a relationship, the SINGULAR meta-attribute of the corresponding RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship in the IRD Schema shall have a value of NO.
- (4) Each attribute in the repeating-attribute-list shall conform to all validation rules defined in the IRD Schema for the corresponding ATTRIBUTE-TYPE.
- (5) The number of values specified in the repeating-attribute-list shall be less than or equal to the value of the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE or RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship.
- (6) A null-mark shall not be specified as a value in a repeating-attribute-list. If it is the only value, a warning shall be issued.

**General Rules**

- (1) This command-clause shall be specified only once for each ATTRIBUTE-TYPE within a command.
- (2) Each attribute in the repeating-attribute-list shall be unique.

**Error and Warning Conditions**

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01148: Attribute not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01128: Invalid attribute or format. Syntax Rule (4) of this command-clause has been violated. This will also apply if a null-mark is one of several values in a repeating-attribute-list (Syntax Rule (6) of this command-clause).
- (4) Error E01132: Too many values specified. Syntax Rule (5) of this command-clause has been violated.

- (5) **Warning W01020:** Attribute does not exist and cannot be erased. A null-mark was specified as the only attribute for a repeating attribute which does not exist.
- (6) **Error E01129:** Attribute specified more than once in command. General Rule (1) of this command-clause has been violated.
- (7) **Error E01149:** Duplicated value. General Rule (2) of this command-clause has been violated.

**NOTES:**

- (1) If Syntax Rule (4) of this command-clause is violated, the error message(s) shall clearly identify which value(s) in the list were in error, and which validation rules were violated for each erroneous value.
- (2) This command-clause corresponds to the new-repeating-meta-attribute IRD-schema-clause in Subsection 6.107.

### 6.5 New Simple Attribute-Group Command-Clause

**Function** To declare a new nonrepeating attribute group within an add-entity-command or an add-relationship-command.

**Format**

```
new-simple-attribute-group-clause ::=
    attribute-group-type-designator           4.6
    =
    attribute-group                           4.5
```

#### Syntax Rules

- (1) The attribute-group-type-designator shall identify an ATTRIBUTE-GROUP-TYPE in the IRD Schema.
- (2) The ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being created or modified by the command.
- (3) If the command operates on an entity, the SINGULAR meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of YES. If the command operates on a relationship, the SINGULAR meta-attribute of the corresponding RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of YES.
- (4) If the positional form of attribute-group is used, each attribute specified shall conform to all validation rules defined in the IRD Schema for the ATTRIBUTE-TYPE at the corresponding GROUP-POSITION within the ATTRIBUTE-GROUP-TYPE. In this form, a component attribute can be left undeclared by leaving the corresponding position within the group either blank or null.
  - (a) If the ATTRIBUTE-GROUP-TYPE contains  $n$  component ATTRIBUTE-TYPEs, then there shall be at most  $n-1$  commas within the attribute-group.



- (b) If there are  $m$  commas, where  $m < n-1$ , the first  $m+1$  component attributes shall be created within the attribute-group based on GROUP-POSITION.
- (5) If the nonpositional form of the attribute-group is used, then a component attribute shall be declared by using the form of the simple-attribute-clause. The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema which is also related in the IRD Schema to the ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator. The value declared for the component attribute shall conform to all validation rules for the corresponding ATTRIBUTE-TYPE. At least one such attribute shall be declared.
- (6) Each component ATTRIBUTE-TYPE of the ATTRIBUTE-GROUP-TYPE which is SIGNIFICANT shall be specified with an attribute which is not null-mark or null.
- (7) Attribute-group-type-designator shall not be the name of an audit attribute-group-type.

#### General Rules

- (1) This command-clause shall be specified only once for each ATTRIBUTE-GROUP-TYPE within a command.

#### Error and Warning Conditions

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01150: Attribute-group not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01133: Missing or unbalanced parentheses. A valid attribute-group format was expected but not recognized.
- (4) Error E01132: Too many values specified. More than one group value was recognized.
- (5) Error E01134: Invalid component attribute. Syntax Rule (4) or (5) has been violated.
- (6) Error E01135: Unrecognized component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator of a simple-attribute-clause did not identify a valid ATTRIBUTE-TYPE in the IRD Schema. See Syntax Rule (5).
- (7) Error E01136: Inconsistent component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator identified an ATTRIBUTE-TYPE in the IRD Schema which was not related to the ATTRIBUTE-GROUP-TYPE identified in this command-clause. See Syntax Rule (5).
- (8) Error E01152: Too many values within group. Syntax Rule (4)(a) of this command-clause has been violated.
- (9) Error E01137: Required attribute missing. Syntax Rule (6) of this command-clause has been violated.
- (10) Error E01131: Repeating attribute specified. Syntax Rule (3) of this command-clause has been violated.

- (11) Error E01216: Declaration of attribute for audit attribute-group-type not allowed. Syntax Rule (7) of this command-clause has been violated.
- (12) Error E01217: Attribute-group-type specified more than once in command. General Rule (1) of this command-clause has been violated.

NOTES:

- (1) The error message(s) shall clearly identify which attributes within the group were in error, and which validation rules were violated for each erroneous value.
- (2) This command-clause corresponds to the new-meta-attribute-group IRD-schema-clause in Subsection 6.108.

## 6.6 New Repeating Attribute-Group Command-Clause

**Function** To declare a new repeating attribute-group within an add-entity-command or an add-relationship-command.

**Format**

```
new-repeating-attribute-group-clause ::=
    attribute-group-type-designator           4.6
    =
    attribute-group-list

attribute-group-list ::=
    attribute-group                           4.5
    [ [,] attribute-group-list]
```

### Syntax Rules

- (1) The attribute-group-type-designator shall identify an ATTRIBUTE-GROUP-TYPE in the IRD Schema.
- (2) The ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being created by the command.
- (3) If the command operates on an entity, the SINGULAR meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of NO. If the command operates on a relationship, the SINGULAR meta-attribute of the corresponding RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of NO.
- (4) If the positional form of attribute-group is used, each component attribute specified shall conform to all validation rules defined in the IRD Schema for the ATTRIBUTE-TYPE at the corresponding GROUP-POSITION within the ATTRIBUTE-GROUP-TYPE. In this form, a component attribute can be left undeclared by leaving the corresponding position within the group either blank or null.
  - (a) If the ATTRIBUTE-GROUP-TYPE contains  $n$  component ATTRIBUTE-TYPEs, then there shall be at most  $n-1$  commas within the attribute-group.

- (b) If there are  $m$  commas, where  $m < n-1$ , the first  $m+1$  component attributes shall be created within the attribute-group based on GROUP-POSITION.
- (5) If the nonpositional form of the attribute-group is used, then a component attribute shall be declared by using the form of the simple-attribute-clause. The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema which is also related in the IRD Schema to the ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator. The value declared for the component attribute shall conform to all validation rules for the corresponding ATTRIBUTE-TYPE. At least one such attribute shall be declared.
- (6) The number of attribute-groups specified shall be less than or equal to the value of the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE or RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship.
- (7) The positional and nonpositional forms of attribute-group can be mixed.
- (8) Within each specified attribute-group, a value shall be declared for each component ATTRIBUTE-TYPE which is significant.

#### General Rules

- (1) This command-clause shall be specified only once for each ATTRIBUTE-GROUP-TYPE within a command.
- (2) Two different attribute-groups shall not have identical sets of significant attributes. (The significant ATTRIBUTE-TYPEs of a repeating ATTRIBUTE-GROUP-TYPE act like an access key for a record. This rule thus states that each attribute-group shall have a unique "key".)

#### Error and Warning Conditions

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01150: Attribute-group not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01133: Missing or unbalanced parentheses. A valid attribute-group format was expected but not recognized.
- (4) Error E01132: Too many values specified. Syntax Rule (6) of this command-clause has been violated.
- (5) Error E01134: Invalid component attribute. Syntax Rule (4) or (5) of this command-clause has been violated.
- (6) Error E01135: Unrecognized component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator of a simple-attribute-clause did not identify a valid ATTRIBUTE-TYPE in the IRD Schema. See Syntax Rule (5) of this command-clause.

- (7) Error E01136: Inconsistent component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator identified an ATTRIBUTE-TYPE in the IRD Schema which was not related to the ATTRIBUTE-GROUP-TYPE identified in this command-clause. See Syntax Rule (5) of this command-clause.
- (8) Error E01152: Too many values within group. Syntax Rule (4)(a) of this command-clause has been violated.
- (9) Error E01138: Required component attribute not specified. Syntax Rule (8) of this command-clause has been violated.
- (10) Error E01139: Attribute-group duplicated. General Rule (2) of this command-clause has been violated.
- (11) Error E01140: Nonrepeating attribute specified. Syntax Rule (3) of this command-clause has been violated.

NOTE: The error message(s) shall clearly identify which attribute-group(s) were erroneous; and, which attributes within each erroneous group were in error, and which validation rules were violated for each erroneous value.

## 6.7 New Text Attribute Command-Clause

**Function** To create an attribute which consists of one or more lines of text.

### Format

```

new-text-attribute-clause ::=
    attribute-type-designator           4.6
    [line-number-defaults]
    =
    string-literal-list

string-literal-list ::=
    string-literal                       4.2
    [ , string-literal-list ]

line-number-defaults ::=
    line-number-defaults-positional-format
    | line-number-defaults-non-positional-format

line-number-defaults-positional-format ::=
    ( start [, increment] )

start ::= unsigned-integer | null       4.2

increment ::= unsigned-integer         4.2

line-number-defaults-non-positional-format ::=
    ( first-default-clause
      [,] [second-default-clause] )

```

first-default-clause ::=	
start-line-number-clause	6.19
line-number-increment-clause	6.20
second-default-clause ::=	
start-line-number-clause	6.19
line-number-increment-clause	6.20

### Syntax Rules

- (1) The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema. Its FORMAT meta-attribute shall have a value of TEXT.
- (2) The ATTRIBUTE-TYPE identified by attribute-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being created or modified by the command.

### General Rules

- (1) This command-clause shall be specified only once for each ATTRIBUTE-TYPE within a command.
- (2) If line-number-defaults is not specified, then start and increment will be assigned implementation-dependent default values.
- (3) Any assigned line number shall not exceed an implementation-dependent maximum value. See subsequent rules for assigning line-numbers.
- (4) The length of a line of text shall be equal to the value of the MAXIMUM-ATTRIBUTE-LENGTH meta-attribute for the attribute-type. If a string-literal exceeds line length, the system shall then subdivide the string-literal into as many lines-of-text as required, assigning a line number to each.
- (5) Lines of text will be assigned line numbers in ascending order, incremented by the increment value. Numbering will begin with the start value. The numbering of lines of text in subsequent string-literals shall begin one increment after the highest line-number assigned in the prior string-literal.
- (6) A null string ("") shall mean the line does not exist. If specified, the system shall bypass the line-number that would have been assigned to this string-literal, and proceed to the next string-literal.
- (7) If only the null-string is specified, the system shall regard the attribute as not having been declared at all.
- (8) If the assignment of line-numbers according to the start and increment values will result in exceeding the implementation-defined maximum line-number value, then all lines shall be stored, and line-numbers shall be reassigned from the last line backwards. Line numbers reassigned in this fashion shall begin with the maximum line number, and decremented by a value of 1, until such a reassigned line number is greater than the currently assigned line number of the prior line.
- (9) If the nonpositional form of line-number-defaults is used, then start-line-number-clause and line-number-increment-clause each shall be specified only once.

### **Error and Warning Conditions**

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01148: Attribute not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01141: Maximum string length exceeded or string-literal not properly delimited.
- (4) Warning W01021: Only null-string specified. Command-Clause ignored. See General Rule (7) of this command-clause.
- (5) Warning W01022: Null string specified. Line-number skipped. See General Rule (6) of this command-clause.
- (6) Warning W01023: Assigned line numbers will exceed maximum. Reassigning line numbers from back, decrementing by 1. See General Rule (8) of this command-clause.
- (7) Error E01151: Duplicated subclause. General Rule (9) of this command-clause has been violated.
- (8) Error E01129: Attribute-type specified more than once in command. General Rule (1) of this command-clause has been violated.

#### **NOTES:**

- (1) The maximum number of lines of text shall be sufficiently large to accommodate practical limitations of human input. This value shall be no less than 32767. An implementation may choose a higher value.
- (2) The length of a line of text shall be implementation-dependent. The implementor shall allow for a line length of at least 72 characters.
- (3) This command-clause corresponds to the new-text-meta-attribute IRD-schema-clause in Subsection 6.109.

### **6.8 Modified Repeating Attribute Command-Clause**

**Function** For an existing entity or relationship:

- (1) To create a new repeating attribute.
- (2) To create new values for an existing repeating attribute.
- (3) To replace a value of an existing repeating attribute by another value.
- (4) To erase one or more values of an existing repeating attribute.
- (5) To erase an existing repeating attribute.

#### **Format**

```
modified-repeating-attribute-clause ::=
    [ all-word-or-new-word ]
```

attribute-type-designator	4.6
[ attribute-occurrence-qualification ]	
=	
repeating-attribute-list	4.5
null-mark	4.1
all-word-or-new-word ::=	
all-word	4.3
new-word	4.3
attribute-occurrence-qualification ::=	
( attribute )	4.5

### Syntax Rules

- (1) The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema.
- (2) The ATTRIBUTE-TYPE identified by attribute-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being modified by the command.
- (3) If the command operates on an entity, the SINGULAR meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship in the IRD Schema shall have a value of NO. If the command operates on a relationship, the SINGULAR meta-attribute of the corresponding RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship in the IRD Schema shall have a value of NO.
- (4) Each attribute in the repeating-attribute-list shall conform to all validation rules defined in the IRD Schema for the corresponding ATTRIBUTE-TYPE.
- (5) The number of values specified in the repeating-attribute-list shall be less than or equal to the value of the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE or RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship.
- (6) Null-mark shall not be specified as one of several attributes within a repeating-attribute-list.
- (7) If all-word or new-word precedes attribute-type-designator, then attribute-occurrence-qualification shall not be specified.
- (8) If attribute-occurrence-qualification is specified, then either null-mark or only one attribute shall be specified after the equals-sign.
- (9) If neither all-word nor new-word is specified and attribute-occurrence-qualification is not specified, all-word shall be assumed.

### General Rules

- (1) For each such ATTRIBUTE-TYPE within a command:
  - (a) If all-word is specified or assumed, no other occurrences of this command-clause shall be allowed for the specified attribute-type-designator.

- (b) If new-word is specified, there can be multiple command-clauses for the specified attribute-type-designator with attribute-occurrence-qualification also specified.
  - (c) All command-clauses for a given attribute-type-designator shall be specified consecutively.
- (2) Each attribute specified after the equals-sign in all occurrences of this command-clause within a command shall be unique.
  - (3) If the ATTRIBUTE-TYPE does not exist in the entity or relationship being modified, then the command-clause shall be treated as a new-repeating-attribute-clause (see Subsection 6.4) from attribute-type-designator on. Thus the command-clause shall be valid even if attribute-type-designator is preceded by all-word or new-word.
  - (4) If attribute-type-designator identifies an attribute-type which exists in the entity or relationship being modified, and attribute-occurrence-qualification is specified, then the attribute specified in attribute-occurrence-qualification shall identify an existing value of the attribute.
  - (5) If attribute-type-designator identifies an attribute-type which exists in the entity or relationship being modified, and attribute-occurrence-qualification is specified, then:
    - (a) A null-mark shall indicate that the attribute specified in attribute-occurrence-qualification is to be erased.
    - (b) An attribute not equal to null-mark shall indicate that the attribute specified in attribute-occurrence-qualification shall be replaced by the attribute after the equals-sign.
    - (c) The attribute specified after the equals-sign shall not be an existing value of the attribute.
  - (6) The result of all such command-clauses for a given attribute-type-designator shall not result in more values being assigned to attribute-type-designator than the value of the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE or RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship.
  - (7) If all-word is specified or assumed, and attribute-type-designator exists within the entity or relationship being modified, a null-mark after the equals-sign shall identify that attribute-type-designator and all its values shall be erased.

### **Error and Warning Conditions**

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01148: Attribute not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01128: Invalid attribute or format. Syntax Rule (3), (4), or (6) of this command-clause has been violated.



- (4) Error E01132: Too many values specified. Syntax Rule (5) or General Rule (6) of this command-clause has been violated.
- (5) Error E01129: Attribute specified more than once in command. General Rule (1) of this command-clause has been violated.
- (6) Error E01149: Duplicated value. General Rule (2) or (5)(c) of this command-clause has been violated.
- (7) Error E01142: Inconsistent command-clause format. Either Syntax Rule (7) or (8) of this command-clause has been violated.
- (8) Error E01143: Current value does not exist. General Rule (4) of this command-clause has been violated.
- (9) Warning W01020: Attribute does not exist and cannot be erased. A null-mark was specified as the only attribute for a repeating attribute which does not exist.
- (10) Warning W01024: All values to be replaced. All-word assumed and attribute-type-designator exists.
- (11) Warning W01025: Attribute is to be erased. Erasing individual attributes results in the entire attribute being erased.

NOTES:

- (1) If Syntax Rule (4) of this command-clause is violated, the error message(s) shall clearly identify which value(s) in the list were in error, and which validation rules were violated for each erroneous value.
- (2) This command-clause corresponds to the modified-repeating-meta-attribute IRD-schema-clause in Subsection 6.110.

**6.9 Modified Simple Attribute-Group Command-Clause**

**Function** For an existing entity or relationship:

- (1) To create a new nonrepeating attribute group.
- (2) To create new nonsignificant attributes within an attribute-group.
- (3) To replace a value of one or more component attributes by other values.
- (4) To erase one or more nonsignificant component attributes of an attribute-group.
- (5) To erase one or more nonsignificant component attributes of an existing attribute-group.
- (6) To erase an existing attribute-group.

**Format**

```

modified-simple-attribute-group-clause ::=
    [ all-word ]           4.3
    attribute-group-type-designator 4.6
    
```

=		
attribute-group		4.5
null-mark		4.1

### Syntax Rules

- (1) The attribute-group-type-designator shall identify an ATTRIBUTE-GROUP-TYPE in the IRD Schema.
- (2) The ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being created or modified by the command.
- (3) If the command operates on an entity, the SINGULAR meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of YES. If the command operates on a relationship, the SINGULAR meta-attribute of the corresponding RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of YES.
- (4) If the positional form of attribute-group is used, each attribute specified shall conform to all validation rules defined in the IRD Schema for the ATTRIBUTE-TYPE at the corresponding GROUP-POSITION within the ATTRIBUTE-GROUP-TYPE. In this form, a component attribute can be left undeclared by leaving the corresponding position within the group either blank or null.
  - (a) If the ATTRIBUTE-GROUP-TYPE contains  $n$  component ATTRIBUTE-TYPEs, then there shall be at most  $n-1$  commas within the attribute-group.
  - (b) If there are  $m$  commas, where  $m < n-1$ , the first  $m+1$  component attributes shall be created within the attribute-group based on GROUP-POSITION.
- (5) If the nonpositional form of the attribute-group is used, then a component attribute shall be declared by using the form of the simple-attribute-clause. The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema which is also related in the IRD Schema to the ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator. The value declared for the component attribute shall conform to all validation rules for the corresponding ATTRIBUTE-TYPE. At least one such attribute shall be declared.
- (6) If attribute-group-type-designator does not exist in the maintained entity or relationship, the command-clause shall be treated as a new-simple-attribute-group-clause from attribute-group-type-designator on. In particular this implies:
  - (a) If all-word is specified it shall be ignored, and
  - (b) Each component ATTRIBUTE-TYPE of the ATTRIBUTE-GROUP-TYPE which is significant shall be specified with an attribute which is not null-mark or null, and
  - (c) Each nonsignificant component ATTRIBUTE-TYPE which has no corresponding value declared shall be regarded as not existing.
- (7) If attribute-group-type-designator does exist in the maintained entity or relationship and all-word is specified, attribute-group shall be regarded as replacing the existing attribute-

group. Attribute-group shall thus be subject to all rules which apply to attribute-group in new-simple-attribute-group-clause. In particular this implies:

- (a) Each component ATTRIBUTE-TYPE of the ATTRIBUTE-GROUP-TYPE which is significant shall be specified with an attribute which is not null-mark or null, and
  - (b) Each nonsignificant component ATTRIBUTE-TYPE which has no corresponding value declared shall be regarded as not existing.
- (8) If attribute-group-type-designator does exist in the maintained entity or relationship and all-word is not specified, only those component attributes which are to be created, modified, and/or erased need be specified. In particular this means:
- (a) That any component attribute which is not declared shall be left unchanged. If the corresponding component attribute is nonexistent within the group and no value is declared, it remains nonexistent.
  - (b) If null-mark is specified for any nonsignificant component attribute, it shall be declared to be erased if it currently exists, or no change if it does not exist.
  - (c) Any component attribute with a value declared other than null-mark shall be declared to be created if it does not exist or its value replaced if it does exist within the group.
- (9) If null-mark is specified after the equals-sign and outside of parentheses, the attribute-group shall be regarded as to be erased.
- (10) Attribute-group-type-designator shall not be the name of an audit attribute-group-type.

### General Rules

- (1) This command-clause shall be specified only once for each ATTRIBUTE-GROUP-TYPE within a command.
- (2) If attribute-group-type-designator is preceded by new-word instead of all-word, it shall be regarded as having too many repetitions.

### Error and Warning Conditions

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01150: Attribute-group not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01133: Missing or unbalanced parentheses. A valid attribute-group format was expected but not recognized.
- (4) Error E01132: Too many values specified. More than one group value was recognized or attribute-group-type-designator was preceded by new-word (General Rule (2)).
- (5) Error E01134: Invalid component attribute. Syntax Rule (4) or (5) has been violated.

- (6) Error E01135: Unrecognized component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator of a simple-attribute-clause did not identify a valid ATTRIBUTE-TYPE in the IRD Schema. See Syntax Rule (5).
- (7) Error E01136: Inconsistent component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator identified an ATTRIBUTE-TYPE in the IRD Schema which was not related to the ATTRIBUTE-GROUP-TYPE identified in this command-clause. See Syntax Rule (5).
- (8) Error E01152: Too many values within group. Syntax Rule (4)(a) has been violated.
- (9) Error E01137: Required attribute missing. Syntax Rule (6)(b), (7)(a), or (8)(b) of this command-clause has been violated.
- (10) Error E01131: Repeating attribute specified. Syntax Rule (3) of this command-clause has been violated.
- (11) Error E01216: Declaration of attribute for audit attribute-group-type not allowed. Syntax Rule (10) has been violated.
- (12) Error E01217: Attribute-group-type specified more than once in command. General Rule (1) of this command-clause has been violated.

NOTES:

- (1) The error message(s) shall clearly identify which attributes within the group were in error, and which validation rules were violated for each erroneous value.
- (2) This command-clause corresponds to the modified-meta-attribute-group IRD-schema-clause in Subsection 6.111.

## 6.10 Modified Repeating Attribute-Group Command-Clause

**Function** For an existing entity or relationship:

- (1) To create a new repeating attribute-group.
- (2) To create new occurrences of an existing attribute-group.
- (3) To erase all or specific occurrences of an attribute-group.
- (4) To create, modify, or erase specific component attributes within a given occurrence of a repeating attribute-group.

**Format**

```
modified-repeating-attribute-group-clause ::=
    [ all-word-or-new-word ]
    attribute-group-type-designator           4.6
    [ group-occurrence-qualification ]
    =
    attribute-group-list
```

group-occurrence-qualification ::=	
attribute-group	4.5
attribute-group-list ::=	
attribute-group	4.5
[ [,] attribute-group-list ]	
all-word-or-new-word ::=	
all-word	4.3
new-word	4.3

### Syntax Rules

- (1) The attribute-group-type-designator shall identify an ATTRIBUTE-GROUP-TYPE in the IRD Schema.
- (2) The ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being modified by the command.
- (3) If the command operates on an entity, the SINGULAR meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of NO. If the command operates on a relationship, the SINGULAR meta-attribute of the corresponding RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE meta-relationship in the IRD Schema shall have a value of NO.
- (4) If the positional form of attribute-group is used, each attribute specified shall conform to all validation rules defined in the IRD Schema for the ATTRIBUTE-TYPE at the corresponding GROUP-POSITION within the ATTRIBUTE-GROUP-TYPE. In this form, a component attribute can be left undeclared by leaving the corresponding position within the group either blank or null.
  - (a) If the ATTRIBUTE-GROUP-TYPE contains  $n$  component ATTRIBUTE-TYPEs, then there shall be at most  $n-1$  commas within the attribute-group.
  - (b) If there are  $m$  commas, where  $m < n-1$ , the first  $m+1$  component attributes will be created within the attribute-group based on GROUP-POSITION.
- (5) If the nonpositional form of the attribute-group is used, then a component attribute shall be declared by using the form of the simple-attribute-clause. The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema which is also related in the IRD Schema to the ATTRIBUTE-GROUP-TYPE identified by attribute-group-type-designator. The value declared for the component attribute shall conform to all validation rules for the corresponding ATTRIBUTE-TYPE. At least one such attribute shall be declared.
- (6) If group-occurrence-qualification is not specified, the number of attribute-groups specified shall be less than or equal to the value of the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE or RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship.
- (7) If group-occurrence-qualification is specified, then either null-mark or only one attribute-group shall be specified after the equals-sign.

- (8) Null-mark shall not be specified as one of several attribute-groups.
- (9) If all-word or new-word precedes attribute-group-type-designator, then group-occurrence-qualification shall not be specified.
- (10) If neither all-word nor new-word is specified and occurrence-qualification is not specified, all-word shall be assumed.

### General Rules

- (1) For each such ATTRIBUTE-GROUP-TYPE within a command:
  - (a) If all-word is specified or assumed, no other occurrences of this command-clause shall be allowed for the specified attribute-group-type-designator.
  - (b) If new-word is specified, there can be additional command-clauses for the specified attribute-group-type-designator with occurrence-qualification also specified.
  - (c) All command-clauses for a given attribute-group-type-designator shall be specified consecutively.
- (2) Each attribute-group specified after the equals-sign in all occurrences of this command-clause for a given attribute-group-type-designator shall identify an attribute-group with a unique set of significant component attributes.
- (3) If attribute-group-type-designator does not exist in the entity or relationship being modified, then the command-clause shall be treated as a new-repeating-attribute-group-clause (see Subsection 6.6) from attribute-group-type-designator on. Thus the command-clause shall be valid even if attribute-group-type-designator is preceded by all-word or new-word.
- (4) If attribute-group-type-designator identifies an attribute-group which exists in the entity or relationship being modified, and group-occurrence-qualification is specified, then the attribute-group specified in occurrence-qualification shall specify all significant attributes within an existing occurrence of the attribute-group. In particular this means:
  - (a) A simple-attribute-clause shall be specified for each significant component ATTRIBUTE-TYPE of the ATTRIBUTE-GROUP-TYPE if positional-form is used for group-occurrence-qualification. If positional-form is used, a non-null value shall be given in each position corresponding to a significant ATTRIBUTE-TYPE.
  - (b) Each value for a significant ATTRIBUTE-TYPE shall conform to the validation rules for that ATTRIBUTE-TYPE.
  - (c) The set of all such attributes in group-occurrence-qualification shall match those of an existing occurrence of the attribute-group.

Note that any values in positional-form or simple-attribute-clauses which do not correspond to a significant component ATTRIBUTE-TYPE shall be ignored in group-occurrence-qualification.

- (5) If attribute-group-type-designator does not exist as an attribute-group in the entity or relationship being modified, then any group-occurrence-qualification shall be invalid.

- (6) If attribute-group-type-designator identifies an attribute-group which exists in the entity or relationship being modified, and group-occurrence-qualification is specified, then:
- (a) A null-mark shall indicate that the occurrence specified in group-occurrence-qualification is to be erased.
  - (b) An attribute-group not equal to null-mark shall indicate that the attribute-group specified in group-occurrence-qualification shall be replaced by the attribute after the equals-sign.
  - (c) Any and/or all significant attributes of the group can be changed. However, the attribute-group specified after the equals-sign shall not produce a set of significant attributes which duplicate an existing attribute-group, except for the attribute-group occurrence identified by group-occurrence-qualification.
- (7) If attribute-group-type-designator does not exist in the maintained entity or relationship, the command-clause shall be treated as a new-repeating-attribute-group-clause from attribute-group-type-designator on. In particular this implies:
- (a) If all-word is specified it shall be ignored, and
  - (b) Each component ATTRIBUTE-TYPE of the ATTRIBUTE-GROUP-TYPE which is significant shall be specified in each attribute-group with an attribute which is not null-mark or null, and
  - (c) Each nonsignificant component ATTRIBUTE-TYPE which has no corresponding value declared shall be regarded as not existing.
- (8) If attribute-group-type-designator does exist in the maintained entity or relationship and all-word is specified, attribute-group shall be regarded as replacing the existing attribute-group. Attribute-group shall thus be subject to all rules which apply to attribute-group in new-simple-attribute-group-clause. In particular this implies:
- (a) Each component ATTRIBUTE-TYPE of the ATTRIBUTE-GROUP-TYPE which is significant shall be specified in each attribute-group with an attribute which is not null-mark or null, and
  - (b) Each nonsignificant component ATTRIBUTE-TYPE which has no corresponding value declared shall be regarded as not existing.
- (9) If attribute-group-type-designator does exist in the maintained entity or relationship and all-word is not specified, only those component attributes which are to be created, modified, and/or erased need be specified. In particular this means:
- (a) That any component attribute which is not declared shall be left unchanged. If the corresponding component attribute is nonexistent within the group and no value is declared, it remains nonexistent.
  - (b) If null-mark is specified for any nonsignificant component attribute, it shall be declared to be erased if it currently exists, or no change if it does not exist.
  - (c) Any component attribute with a value declared other than null-mark shall be declared to be created if it does not exist or its value replaced if it does exist within the group.

- (10) The result of all such command-clauses for a given attribute-group-type-designator shall not result in more values being assigned to attribute-group-type-designator than the value of the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute of the corresponding ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE or RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE meta-relationship.
- (11) If all-word is specified or assumed, and attribute-group-type-designator exists within the entity or relationship being modified, a null-mark after the equals-sign shall identify that all occurrences of attribute-group-type-designator are to be erased.

### **Error and Warning Conditions**

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01148: Attribute not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01133: Missing or unbalanced parentheses. A valid attribute-group format was expected but not recognized.
- (4) Error E01132: Too many values specified. Either Syntax Rule (6) or (7), or General Rule (10) of this command-clause has been violated.
- (5) Error E01134: Invalid component attribute. Syntax Rule (4) or (5) has been violated.
- (6) Error E01152: Too many values within group. Syntax Rule (4)(a) has been violated.
- (7) Error E01135: Unrecognized component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator of a simple-attribute-clause did not identify a valid ATTRIBUTE-TYPE in the IRD Schema. See Syntax Rule (5).
- (8) Error E01136: Inconsistent component attribute-type. The nonpositional form of the attribute-group was used, and the attribute-type-designator identified an ATTRIBUTE-TYPE in the IRD Schema which was not related to the ATTRIBUTE-GROUP-TYPE identified in this command-clause. See Syntax Rule (5).
- (9) Error E01139: Attribute-group duplicated. General Rule (2), (6)(c), or (7)(c) has been violated.
- (10) Error E01142: Inconsistent command-clause format. Syntax Rule (9) of this command-clause has been violated.
- (11) Error E01144: Current attribute-group does not exist. General Rule (4) of this command-clause has been violated.
- (12) Error E01137: Required attribute missing. Syntax Rule (6)(b), (7)(a), or (8)(b) of this command-clause has been violated.
- (13) Error E01145: Incomplete or invalid qualification. General Rule (4)(a) or (4)(b) has been violated.



- (14) Error E01146: Attribute-group occurrence does not exist. General Rule (4)(c) has been violated, but there are other occurrences which do exist, and the specified occurrence is not to be erased.
- (15) Error E01140: Nonrepeating attribute specified. Syntax Rule (3) of this command-clause has been violated.
- (16) Warning W01026: Attribute-group cannot be erased. Either:
  - (a) Erasure of all occurrences has been requested. No occurrences of attribute-group-type-designator exist within the maintained entity or relationship. Or,
  - (b) General Rule (4)(c) has been violated but the specified occurrence is to be erased.
- (17) Warning W01024: All values to be replaced. All-word assumed and attribute-group-type-designator exists.
- (18) Warning W01027: Attribute-group is to be erased. Erasing individual attribute-group occurrences results in the entire attribute being erased.

NOTE: If Syntax Rule (4) or (5) of this command-clause is violated, the error message(s) shall clearly identify which attribute(s) or simple-attribute-clause(s) in which group is in error, and which validation rules were violated in each case.

### 6.11 Modified Text Attribute Command-Clause

**Function** For an existing entity or relationship:

- (1) To create a text attribute.
- (2) To erase a text attribute.
- (3) To create, modify, or erase any group of one or more lines of text.

**Format**

```

modified-text-attribute-clause ::=
    [ all-word ]                4.3
    attribute-type-designator    4.6
    [ line-qualification ]
    =
    line-group | null-mark
    { subsequent-lines-subclause }

line-qualification ::=
    line-range-clause           6.21
    | line-number-defaults

line-group ::=
    string-literal              4.2
    [ , line-group ]
    
```

subsequent-lines-subclause ::=	
lines-word	4.3
=	
line-range-clause	6.21
line-group   null-mark	
line-number-defaults ::=	
line-number-defaults-positional-format	
line-number-defaults-non-positional-format	
line-number-defaults-positional-format ::=	
( start [, increment ] )	
start ::= unsigned-integer   null	4.2
increment ::= unsigned-integer	4.2
line-number-defaults-non-positional-format ::=	
( first-default-clause [, ]	
second-default-clause )	
first-default-clause ::=	
start-line-number-clause	6.19
line-number-increment-clause	6.20
second-default-clause ::=	
start-line-number-clause	6.19
line-number-increment-clause	6.20

### Syntax Rules

- (1) The attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema. Its FORMAT meta-attribute shall have a value of TEXT.
- (2) The ATTRIBUTE-TYPE identified by attribute-type-designator shall be related in the IRD Schema to the ENTITY-TYPE or RELATIONSHIP-TYPE which is being created or modified by the command.
- (3) If all-word is specified, line-range-clause is not allowed.
- (4) If the nonpositional form of line-number-defaults is used, then start-line-number-clause and line-number-increment-clause each shall be specified only once.

### General Rules

- (1) This command-clause can be specified any number of times in a command for any given attribute-type-designator. If specified a multiple number of times within a command, each pair of these command-clause shall be separated by a text-attribute-resequence-clause (see Subsection 6.29) for the same text-attribute-type-designator.
- (2) If attribute-type-designator does not exist within the maintained entity or relationship, all-word shall be assumed if not declared. Line-range-clause is not allowed.

- (3) If attribute-type-designator exists within the maintained entity or relationship, line-number-defaults is not allowed.
- (4) If all-word is specified or assumed and line-number-defaults is not specified, then start and increment will be assigned implementation-dependent default values.
- (5) The length of a line of text shall be equal to the value of the MAXIMUM-ATTRIBUTE-LENGTH meta-attribute for the attribute-type. If a string-literal exceeds line length, system shall then subdivide the string-literal into as many lines-of-text as required, assigning a line number to each.
- (6) Any assigned or specified line number shall not exceed an implementation-dependent maximum value. See subsequent rules for assigning line-numbers.
- (7) If all-word is specified or assumed, then this command-clause shall be identical in format to new-text-attribute-clause from attribute-type-designator on, except that a null-mark can be specified after the equals-sign. In particular, this means that subsequent-lines-subclause shall not be specified. The following rules apply in this case:
  - (a) If a null-mark is specified after the equals-sign, the attribute shall be designated to be erased. If it does not exist, a warning condition shall exist.
  - (b) If a line-group is specified after the equals-sign, the line-group shall replace all existing lines, if any. This line group shall be processed exactly as the string-literals of new-text-attribute-clause. See Subsection 6.7 for details.
- (8) If a line-number-range is specified after attribute-type-designator, then attribute-type-designator shall exist in the maintained entity or relationship. The following rules shall apply in this case and to all subsequent-lines-subclauses:
  - (a) If one line number is specified in line-range-clause, then either null-mark or one string-literal no longer than line-length shall be specified to the right of the equals-sign. In this case:
    - (i) A null-mark or null-string shall identify that the line is to be deleted. If it does not exist, a warning condition shall exist.
    - (ii) If the specified line-number does not exist, the string-literal shall be designated to become a new line of text.
    - (iii) If the specified line-number exists, the string-literal shall be designated to replace the existing line in its entirety.
  - (b) If two line-numbers are specified in the line-range-clause, then all existing lines with numbers greater-than or equal-to line-number-1 and less-than or equal-to line-number-2 shall be designated to be replaced or erased. In particular, it is not necessary that either line-number-1 or line-number-2 of the line-range-clause match an existing line-number of attribute-type-designator.
    - (i) If null-mark or null-string is specified after the equals-sign, all lines within the range shall be to be deleted. If no lines exist within the specified range, a warning condition shall exist.

- (ii) If a line-group is specified to the right of the equals-sign, then the line-group shall identify lines which shall replace the existing lines, if any, within the specified range.
- (iii) The line-group shall not specify more lines than can be placed within the range.
- (iv) Line-numbers shall be assigned as follows:
  - (A) The total number of lines within the line-group shall be determined. Let  $N$  denote this number.
  - (B) Let  $L$  and  $H$  represent the line-number-1 and line-number-2 of the line-range-clause, respectively.
  - (C) The increment-value,  $I$ , shall be computed as follows:
$$I = \text{greatest-integer-of } ( (H-L)/N ).$$
  - (D) Line numbers shall then be assigned to individual lines, starting with  $L$  and incremented by  $I$ .

(9) All line ranges specified in this command-clause shall be disjoint.

#### **Error and Warning Conditions**

- (1) Error E01147: Invalid or unrecognized attribute-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01148: Attribute not consistent with entity or relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01141: Maximum string length exceeded or string-literal not properly delimited.
- (4) Error E01153: Inconsistent qualification. Syntax Rule (3) of this command-clause has been violated.
- (5) Error E01108: Duplicated command-clause. General Rule (1) of this command-clause has been violated.
- (6) Error E01151: Duplicated subclause. Syntax Rule (4) of this command-clause has been violated.
- (7) Error E01154: Subsequent lines subclause not allowed. See General Rule (7) of this command-clause.
- (8) Error E01155: String-literal exceeds line length. See General Rule (8)(a) of this command-clause.
- (9) Error E01156: Line-group exceeds specified range. See General Rule (8)(b) of this command-clause.
- (10) Error E01157: Line or range of lines duplicated. General Rule (9) of this command-clause has been violated.

- (11) Warning W01020: Attribute does not exist and cannot be erased. See General Rule (7)(a) of this command-clause.
- (12) Warning W01028: Specified range contains no lines. Lines cannot be deleted. See General Rules (8)(a)(i) and (8)(b)(i) of this command-clause.

NOTES:

- (1) Command-Clause 6.7 provides a description of rules and error conditions which apply if all-word is assumed or specified.
- (2) This command-clause corresponds to the modified-text-meta-attribute IRD-schema-clause in Subsection 6.112.

**6.12 Entity-Access-Name Selection Command-Clause**

**Function** To retrieve entities by entity-access-name or a substring within entity-access-name.

**Format**

```

entity-access-name-selection-clause ::=
    entities-word                4.3
    [ with-word ]                4.3
    access-name-word             4.3
    =
    access-name-scan-pattern-list

access-name-scan-pattern-list ::=
    access-name-pattern
    [ , access-name-scan-pattern-list ]

access-name-pattern ::=
    name-scan-pattern           4.4
    | entity-access-name        4.4
    
```

**Syntax Rules**

- (1) Entity-access-names and name-scan-patterns can be mixed in the list.
- (2) Each name-scan-mask in each name-scan-pattern shall not exceed the maximum number of characters which is allowed for the corresponding part of an entity-access-name.
- (3) If no substitution character is given in the variation-name part of the name-scan-mask, then it shall specify a valid variation-name.

**General Rules**

- (1) If no substitution character is specified, the name-scan-pattern shall be interpreted as an entity-access-name of an entity which exists in the IRD.

### Error and Warning Conditions

- (1) Error E01005: Invalid name format. One or more of the following situations has occurred:
  - (a) The assigned-access-name part contained an invalid character.
  - (b) The variation-name part contained an invalid character.
  - (c) Parentheses were not balanced in the version-identifier.
  - (d) The colon was missing in the version-identifier.
  - (e) The length of the entity's assigned-access-name or variation-name exceeded the maximum allowed.
  - (f) The revision-number part of the entity-access-name did not contain an unsigned integer or an asterisk.
- (2) Error E01009: Entity does not exist or is not visible. No substitution-character was specified and the specified entity-access-name did not exist or was not visible.
- (3) Error E01158: Variation-name does not exist. No substitution-character was specified in the variation-name part of the entity-access-name or name-scan-pattern.

### 6.17 Order Command-Clause

**Function** For a relationship of a relationship-type which is sequenced but does not have a sequencing attribute-type:

- (1) To identify the relative position of a new or existing relationship.
- (2) To provide a new relative position for an existing relationship as part of a modify-relationship-command.

### Format

```
order-clause ::=
    order-word           4.3
    =
    unsigned-integer    4.2
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

### 6.18 Relationship-Type Qualification Command-Clause

**Function** To identify the relationship-type(s) which will qualify subsequent restriction-clauses.

**Format**

```

relationship-type-qualification-clause ::=
    for-word                               4.3
    [ relationships-word                   4.3
      of-word ]                             4.3
    relationship-type-list

relationship-type-list ::=
    relationship-type-designator           4.6
    [ , relationship-type-list ]

```

**Syntax Rules**

- (1) Each relationship-type-designator shall identify a RELATIONSHIP-TYPE meta-entity in the IRD Schema.

**General Rules**

- (1) For-word denotes all qualification. A qualification-clause is either an entity-type qualification-clause or a relationship-type qualification-clause. Attribute-group-restriction-clause also begins with for-word.
- (2) If relationships-word and of-word is not specified after for-word, then the command-clause shall be determined by the first token after for-word.

**Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. An IRD-schema-descriptor was expected. Designator is not a meta-entity in the IRD Schema.
- (2) Error E01105: Invalid relationship-type-designator or inverse. Relationships-word and of-word followed for-word, or the first designator in the list identified a valid RELATIONSHIP-TYPE. Designator in list does not identify a RELATIONSHIP-TYPE meta-entity in the IRD Schema.
- (3) Error E01127: Improperly qualified command-clause. IRD-schema-descriptor is not allowed for qualification. Only an entity-type-designator, relationship-type-designator or attribute-group-type-designator is allowed.

NOTE: This command-clause corresponds to the meta-relationship-type-qualification IRD-schema-clause in Subsection 6.121.

**6.19 Start Line Number Command-Clause**

**Function** For a new text attribute, to identify the initial line number to be assigned. For an existing text attribute, to identify the new initial line number given as part of resequencing the line numbers.

**Format**

```

start-line-number-clause ::=
    start-word                               4.3

```

=  
starting-line-number  
starting-line-number ::= unsigned-integer 4.2

**Syntax Rules**

- (1) Starting-line-number shall be less than the maximum number of lines for a text attribute.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01159: Invalid line number. Line number is not an integer or exceeds maximum allowed value.

**6.20 Line Number Increment Command-Clause**

**Function** To identify the number by which line-numbers shall be incremented when they are initially assigned or reassigned.

**Format**

line-number-increment-clause ::=  
increment-word 4.3  
=  
increment-value  
increment-value ::= unsigned-integer 4.2

**Syntax Rules**

- (1) Increment-value shall not exceed the implementation-defined maximum number of lines of text for a text attribute.
- (2) Increment-value shall not be zero.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01160: Invalid increment value. Increment value is not an unsigned integer, exceeds maximum value, or is zero.

**6.21 Line Range Command-Clause**

**Function** To restrict the maintenance or display of lines of text to a specific line or range of lines.



**Format**

```

line-range-clause ::=
    ( line-number-1
      [ through-word
        line-number-2 ] ) 4.3

line-number-1 ::=
    unsigned-integer      4.2
    | first-word          4.3
    | last-word           4.3

line-number-2 ::=
    unsigned-integer      4.2
    | first-word          4.3
    | last-word           4.3
    | maximum-word        4.3

```

**Syntax Rules**

- (1) If only line-number-1 is specified, then line-number-2 shall be assumed to equal line-number-1.
- (2) If line-number-2 is also specified, then it shall be greater than or equal-to line-number-1.
- (3) Last-word shall always be greater-than or equal-to first-word.

**General Rules**

- (1) First-word is equivalent to specifying the line-number of the first line of text within an existing attribute. Thus it is possible to insert lines of text before the first line of text by specifying:

(*n* THROUGH FIRST)

where *n* is any unsigned integer lower than the first line number. In this case, the first line number shall not be part of the range.

- (2) Last-word is equivalent to specifying the line-number of the last line of text within an existing attribute. Thus it is possible to insert lines of text after the last line of text by specifying:

(LAST THROUGH *m*)

where *m* is any unsigned integer greater than the last line number or maximum-word. In this case, the last line number shall not be part of the range.

If maximum-word shall be specified in this case, or if the computed increment-value exceeds the default increment value, line numbers shall be assigned as multiples of the default increment value, starting with the lowest multiple of increment which is greater than the current last line number. Otherwise, line number increments shall be assigned as specified in Subsection 6.11.

- (3) The range (FIRST THROUGH LAST) shall include the boundaries.

### Error and Warning Conditions

- (1) Error E01159: Invalid line number.

### 6.22 New Entity Attributes Command-Clause

**Function** To identify those attributes and/or attribute-groups which are to be created within an entity by an add-entity-command.

#### Format

```
new-entity-attributes-clause ::=
    with-word                               4.3
    [ attributes-word ]                     4.3
    new-attribute-or-attribute-group-list

new-attribute-or-attribute-group-list ::=
    new-attribute-or-attribute-group
    [ , new-attribute-or-attribute-group-list ]

new-attribute-or-attribute-group ::=
    simple-attribute-clause                 6.3
    | new-repeating-attribute-clause         6.4
    | new-simple-attribute-group-clause     6.5
    | new-repeating-attribute-group-clause  6.6
    | new-text-attribute-clause             6.7
```

**Syntax Rules** None.

**General Rules** None.

### Error and Warning Conditions

- (1) Error E01044: Invalid, missing, or misplaced command-clause. With-word expected and not recognized or attributes-word misspelled.

NOTE: This command-clause corresponds to the new-meta-attributes IRD-schema-clause in Subsection 6.103.

### 6.23 Modified Entity Attributes Command-Clause

**Function** To identify those attributes and/or attribute-groups which can be modified, created, or erased within an entity by a modify-entity-command.

#### Format

```
modified-entity-attributes-clause ::=
    with-word                               4.3
    [ attributes-word ]                     4.3
    modified-attribute-or-attribute-group-list
```

```

modified-attribute-or-attribute-group-list ::=
    modified-attribute-or-attribute-group
    [ , modified-attribute-or-attribute-group-list ]

modified-attribute-or-attribute-group ::=
    simple-attribute-clause                6.3
    | modified-repeating-attribute-clause   6.8
    | modified-simple-attribute-group-clause 6.9
    | modified-repeating-attribute-group-clause 6.10
    | modified-text-attribute-clause        6.11
    | text-attribute-resequence-clause      6.29
    
```

**Syntax Rules**    None.

**General Rules**    None.

**Error and Warning Conditions**

- (1) Error E01044: Invalid, missing, or misplaced command-clause. With-word expected and not recognized or attributes-word misspelled.

NOTE: This command-clause corresponds to the modified-meta-attributes IRD-schema-clause in Subsection 6.104.

**6.24 Relationship Identification Command-Clause**

**Function**    To identify an existing relationship in the IRD.

**Format**

```

relationship-identification-clause ::=
    entity-1-name
    relationship-type-or-relationship-class-type
    entity-2-name
    [ order-clause-or-sequence-attribute ]

entity-1-name ::= entity-access-name           4.4

entity-2-name ::= entity-access-name           4.4

relationship-type-or-relationship-class-type ::=
    relationship-type-designator               4.6
    | relationship-class-type-designator       4.6

order-clause-or-sequence-attribute ::=
    order-clause                               6.17
    | sequence-attribute

sequence-attribute ::= simple-attribute-clause 6.3
    
```

**Syntax Rules**

- (1) Relationship-type-or-relationship-class-type shall be the name of a RELATIONSHIP-TYPE or RELATIONSHIP-CLASS-TYPE meta-entity in the IRD Schema.

## General Rules

- (1) Entity-1-name shall be the entity-access-name of an entity which exists in the IRD.
- (2) If the relationship-class-type is specified, entity-2-name shall be the entity-access-name of an entity which exists in the IRD.
- (3) If the RELATIONSHIP-TYPE identified either by the relationship-type-designator or the combination of the relationship-class-type-designator and the respective ENTITY-TYPEs of entity-1-name and entity-2-name is an ordered relationship-type, then the order-clause or sequence-attribute shall be specified.
- (4) If the RELATIONSHIP-TYPE identified as above is not ordered, then neither the order-clause nor any attribute shall be specified.
- (5) There shall exist a relationship of the appropriate type between entity-1-name and entity-2-name.
- (6) If the relationship is of a RELATIONSHIP-TYPE which is ordered, there shall exist a relationship of that type between entity-1-name and entity-2-name with an order value or sequence-attribute as specified.

## Error and Warning Conditions

- (1) Error E01171: Invalid relationship-type-designator or relationship-class-type-designator. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01161: Relationship and entity-1-name do not exist. General Rule (1) of this command-clause has been violated.
- (3) Error E01162: Relationship from entity-1-name to entity-2-name does not exist. General Rule (2) of this command-clause has been violated.
- (4) Error E01172: Missing order-clause or sequence attribute, or incorrect sequence attribute. General Rule (3) of this command-clause has been violated.
- (5) Error E01044: Invalid, missing or misplaced command-clause. General Rule (4) of this command-clause has been violated.
- (6) Error E01018: Relationship does not exist. General Rule (5) of this command-clause has been violated.
- (7) Error E01163: Incorrect ORDER or sequence-attribute. General Rule (6) of this command-clause has been violated.

## 6.25 New Relationship Identification Command-Clause

**Function** To identify a relationship which is to be created.

### Format

```
new-relationship-identification-clause ::=
    new-relationship-between-existing-entities
```

relate-existing-entity-to-new-entity	
new-relationship-between-existing-entities ::=	
entity-1-name	
relationship-type-or-relationship-class-type	
entity-2-name	
[ order-clause-or-sequence-attribute ]	
relate-existing-entity-to-new-entity ::=	
entity-1-name	
relationship-type-or-relationship-class-type	
new-word	4.3
[ entity-2-entity-type-designator ]	
entity-2-name-or-null-mark	
[ order-clause-or-sequence-attribute ]	
entity-1-name ::= entity-access-name	4.4
entity-2-name ::= entity-access-name	4.4
relationship-type-or-relationship-class-type ::=	
relationship-type-designator	4.6
relationship-class-type-designator	4.6
order-clause-or-sequence-attribute ::=	
order-clause	6.17
sequence-attribute	
sequence-attribute ::=	
simple-attribute-clause	6.3
entity-2-entity-type-designator ::=	
entity-type-designator	4.6
entity-2-name-or-null-mark ::=	
entity-2-name	
null-mark	4.1

### Syntax Rules

- (1) The relationship-type-or-relationship-class-type shall be the name of a RELATIONSHIP-TYPE or RELATIONSHIP-CLASS-TYPE meta-entity in the IRD Schema, in the CONTROLLED IRD Schema life-cycle-phase.
- (2) If new-word is specified after relationship-type-or-relationship-class-type, and a valid RELATIONSHIP-CLASS-TYPE is identified, then entity-2-entity-type-designator shall be required.
- (3) If new-word is specified after relationship-type-or-relationship-class-type, and a valid RELATIONSHIP-TYPE is identified, then entity-2-entity-type-designator shall be optional.
- (4) If new-word is specified after relationship-type-or-relationship-class-type, then entity-2-entity-type-designator shall identify an ENTITY-TYPE in the IRD Schema, in the CONTROLLED IRD Schema life-cycle-phase.

- (5) If a RELATIONSHIP-TYPE is specified, the entity-type of entity-1-name shall be related in the IRD Schema to the identified RELATIONSHIP-TYPE meta-entity with a POSITION meta-attribute of 1. If in addition new-word is not specified, then the entity-type of entity-2-name shall be related in the IRD Schema to the identified RELATIONSHIP-TYPE with a POSITION meta-attribute of 2.

If a RELATIONSHIP-CLASS-TYPE is specified, then there shall exist in the IRD Schema a RELATIONSHIP-TYPE which:

- (a) Is related to the identified RELATIONSHIP-CLASS-TYPE, and
  - (b) The entity-type of entity-1-name is related to the RELATIONSHIP-TYPE in the IRD Schema with a POSITION meta-attribute of 1, and
  - (c) entity-2-entity-type-designator shall be related to the RELATIONSHIP-TYPE in the IRD Schema with a POSITION meta-attribute of 2.
- (6) If new-word is specified and the entity-type for entity-2 as identified by either the relationship-type-designator or entity-2-entity-type-designator is defined in the IRD Schema as having system-generated-names, then either null-mark or entity-2-name can be specified.
- (7) If new-word and the entity-type for entity-2 as identified by either the relationship-type-designator or entity-2-entity-type-designator is defined in the IRD Schema as not having system-generated-names, then null-mark shall not be specified.

### General Rules

- (1) If new-word is not specified, then entity-1-name and entity-2-name shall be entity-access-names of existing entities in the IRD.
- (2) If a valid RELATIONSHIP-CLASS-TYPE is specified, and new-word is not specified, then there shall be a RELATIONSHIP-TYPE in the IRD Schema which:
  - (a) Is related in the IRD Schema to the RELATIONSHIP-CLASS-TYPE, and
  - (b) Is related in the IRD Schema to the ENTITY-TYPE of entity-1-name, and
  - (c) Is related in the IRD Schema to the ENTITY-TYPE of entity-2-name.
- (3) If new-word is specified, then entity-1-name shall be the entity-access-name of existing entity in the IRD, and entity-2-name shall not be the entity-access-name of an existing entity in the IRD.
- (4) If a valid RELATIONSHIP-CLASS-TYPE is specified, and new-word is specified, then there shall be a RELATIONSHIP-TYPE in the IRD Schema which:
  - (a) Is related in the IRD Schema to the RELATIONSHIP-CLASS-TYPE, and
  - (b) Is related in the IRD Schema to the ENTITY-TYPE of entity-1-name, and
  - (c) Is related in the IRD Schema to the ENTITY-TYPE identified by entity-2-entity-type-designator.

- (5) If the identified RELATIONSHIP-TYPE is defined in the IRD Schema as not ordered, then sequence-attribute or order-clause shall not be specified.
- (6) If the identified RELATIONSHIP-TYPE is defined in the IRD Schema as ordered, then the order-clause shall be specified.
- (7) If the identified RELATIONSHIP-TYPE is defined in the IRD Schema as ordered by a sequence-attribute-type, then the sequence-attribute shall be specified.
- (8) If the identified RELATIONSHIP-TYPE is not ordered, then there shall not exist any relationship in the IRD of the identified RELATIONSHIP-TYPE with entity-1-name as the first entity of the relationship and entity-2-name as the second entity of the relationship.
- (9) If the identified RELATIONSHIP-TYPE is ordered, then there shall exist no relationship in the IRD of the identified RELATIONSHIP-TYPE with entity-1-name as the first entity of the relationship and with the order-value or sequence-attribute as specified.

#### **Error and Warning Conditions**

- (1) Error E01171: Invalid relationship-type-designator or relationship-class-type-designator. Syntax Rule (1) has been violated.
- (2) Error E01002: Invalid or unrecognized entity-type. Syntax Rule (2) or (4) has been violated.
- (3) Error E01173: Specified entity-type is inconsistent. Syntax Rule (5) or General Rule (2) or (4) has been violated.
- (4) Error E01009: Entity does not exist or is not visible. Either General Rule (1) or General Rule (3) has been violated.
- (5) Error E01044: Invalid, missing, or misplaced command-clause. General Rule (5) has been violated.
- (6) Error E01172: Missing order-clause or sequence attribute, or incorrect sequence attribute. General Rule (6) or (7) has been violated.
- (7) Error E01164: Duplicated relationship. General Rule (8) has been violated.
- (8) Error E01017: Relationship already exists. General Rule (9), and possibly General Rule (8), has been violated.
- (9) Error E01004: User designated entity-access-name required. Syntax Rule (7) has been violated.

#### **6.26 Order Restriction Command-Clause**

**Function** To restrict the selection of entities and/or relationships based upon a test of the value of the order of a sequenced relationship-type.

**Format**

order-restriction-clause ::=	
order-word	4.3
relational-operator	4.3
unsigned-integer	4.2

**Syntax Rules**

- (1) This command-clause shall be governed by a relationship-type-qualification-clause which identifies a sequenced RELATIONSHIP-TYPE for which no special sequencing ATTRIBUTE-TYPE has been designated.

**General Rules**     None.

**Error and Warning Conditions**

- (1) Error E01174: Inconsistent IRD-schema-descriptors. Syntax Rule (1) of this command-clause has been violated.

**6.27 Entity-Descriptive-Name Declaration Command-Clause**

**Function**     To declare the entity-descriptive-name for an entity.

**Format**

entity-descriptive-name-declaration-clause ::=	
entity-word	4.3
descriptive-name-word	4.3
= assigned-descriptive-name	4.4

**Syntax Rules**     None.

**General Rules**

For the following General Rules:

Let *X* and *Y* denote different assigned-access-names for entities.  
Let *V<sub>i</sub>* and *V<sub>j</sub>* denote variation-names.  
Let *n<sub>i</sub>* and *n<sub>j</sub>* denote revision-numbers.  
Let *D1* and *D2* denote assigned-descriptive-names for entities.

General rules are now stated, as follows:

- (1) If *X(V<sub>i</sub>:n<sub>i</sub>)* has an assigned-descriptive-name of *D1*, then so shall all *X(V<sub>j</sub>:n<sub>j</sub>)* for all *V<sub>i</sub>, V<sub>j</sub>, n<sub>i</sub>*, and *n<sub>j</sub>*.
- (2) If *X(V<sub>i</sub>:n<sub>i</sub>)* has assigned-descriptive-name *D1*, and *Y(V<sub>j</sub>:n<sub>j</sub>)* has assigned-descriptive-name *D2*, then *D1* < > *D2*.
- (3) If *X(V<sub>i</sub>:n<sub>i</sub>)* has assigned-descriptive-name *D1*, then *D1* < > *Y*.
- (4) Assigned-descriptive-name shall conform to all the rules for an assigned-descriptive-name.



**Error and Warning Conditions**

- (1) Error E01031: Entity-descriptive-name conflict. An attempt was made to create an entity with an assigned-descriptive-name which would have violated General Rule (2) of this command-clause.
- (2) Error E01032: Invalid entity-descriptive-name. General Rule (4) has been violated.

NOTE: This command-clause corresponds to the meta-entity-descriptive-name IRD-schema-clause in Subsection 6.126.

**6.28 New Version Command-Clause**

**Function** To assign a version-identifier to a new entity which has been created by a modify-entity-command, modify-entity-life-cycle-phase-command, or copy-entity-command.

**Format**

```

new-version-clause ::=
    new-word           4.3
    entity-word        4.3
    version-word       4.3
    [ = version-identifier ] 4.4

```

**Syntax Rules** None.

**General Rules**

- (1) If version-identifier is not specified, then the system shall assign a new version-identifier with the same variation-name as the specified entity. The revision-number shall be 1 greater than the highest revision number associated with the variation-name and entity-access-name.
- (2) If a revision-number is specified, it shall be different than any revision-number for any entity having the specified assigned-access-name and the specified or defaulted variation-name.

**Error and Warning Conditions**

- (1) Error E01006: Invalid version-identifier. A version-identifier was expected, but the token did not conform to version-identifier format.

NOTE: This command-clause corresponds to the new-meta-entity-version IRD-schema-clause in Subsection 6.127.

**6.29 Text Attribute Resequencing Command-Clause**

**Function** To specify how to resequence all or a specified range of lines of text of a text attribute.

**Format**

```

text-attribute-resequence-clause ::=
    text-attribute-type-designator

```

[ line-range-clause ]	6.21
=	
resequenced-word	4.3
[ start-line-number-clause ]	6.19
[ line-number-increment-clause ]	6.20
text-attribute-type-designator ::=	
attribute-type-designator	4.6

### Syntax Rules

- (1) Text-attribute-type-designator shall be the name of an ATTRIBUTE-TYPE meta-entity in the IRD Schema with a FORMAT meta-attribute equal to TEXT.

### General Rules

- (1) If no line-range-clause is specified, the range "(FIRST THROUGH LAST)" shall be assumed.
- (2) If start-line-number-clause is not specified, then:
- If all line numbers are to be resequenced, then this command-clause shall have the same default as the starting line number for a new text attribute.
  - If a line range is specified, the first line within the range shall retain the same line number.
- (3) If start-line-number-clause is specified, then resequencing of line numbers shall begin with the first line of text within the specified or assumed range. The line number assigned to this first line of text shall be as specified in the start-line-number-clause.
- (4) If the specified or assumed line-range ends with LAST or MAXIMUM, then each line-number after the first line-number within the range shall have a value equal to the prior line number plus the (specified or defaulted) increment value. Line number processing from the initial value of the range shall then be the same as for a new text attribute.
- (5) If the end of the range is less than the last line number within the text attribute, then:

Letting ( $L$  THROUGH  $H$ ) represent the specified range,  
 let  $N$  represent the number of lines in the range ( $L$  THROUGH  $H$ ),  
 let  $I$  denote the increment value, and  
 let  $S$  be the specified or assumed value for starting line number.

- The range ( $S$  THROUGH  $S+N$ ) shall not include any existing line with a line number outside the range ( $L$  THROUGH  $H$ ). An error condition shall be issued.
- If line-number-increment-clause is specified, the value of  $I$  shall be as specified if the range ( $S$  THROUGH  $S+(N-1)*I$ ) does not include any line number outside the range ( $L$  THROUGH  $H$ ).
- If line-number-increment-clause is not specified, the value of  $I$  shall be the same as the default for a new text attribute if the range ( $S$  THROUGH  $S+(N-1)*I$ ) does not include any line number outside the range ( $L$  THROUGH  $H$ ).

- (d) If rule (5)(b) or (5)(c) is violated, then  $I$  shall be computed as follows:  $I = \text{greatest-integer-of } ((H-L)/N)$ . In this case a warning condition shall be issued.

**Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01165: Resequencing line number conflict. General Rule (5)(b) of this command-clause has been violated.
- (3) Warning W01029: Defaulted or specified increment value has been overridden. See General Rule (5)(e) of this command-clause.

NOTE: This command-clause corresponds to the text-meta-attribute-resequence IRD-schema-clause in Subsection 6.125.

**6.30 Quality Indicator Designation Command-Clause**

**Function** To designate a specified entity as being of a specified level of quality.

**Format**

```

quality-indicator-designation-clause ::=
    quality-word                               4.3
    =
    quality-indicator-designator              4.6
    
```

**Syntax Rules**

- (1) The specified quality-indicator-designator shall be the meta-entity-access-name or meta-entity-substitute-name of a meta-entity of type QUALITY-INDICATOR.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01175: Invalid quality-indicator-designator. Syntax Rule (1) has been violated.

**6.31 Quality Indicator Restriction Command-Clause**

**Function** To restrict selection of an entity based its designated quality.

**Format**

```

quality-indicator-restriction-clause ::=
    quality-word                               4.3
    equal-or-not-equal
    quality-indicator-designator              4.6

equal-or-not-equal ::=
    equals                                     4.3
    | not-equal                               4.3
    
```

### Syntax Rules

- (1) The specified quality-indicator-designator shall be the meta-entity-access-name or meta-entity-substitute-name of a meta-entity of type QUALITY-INDICATOR.

**General Rules** None.

### Error and Warning Conditions

- (1) Error E01175: Invalid quality-indicator-designator. Syntax Rule (1) has been violated.

## 6.32 With Relationships Command-Clause

**Function** To specify that new relationships are to be created when a new entity is created by a copy-entity-command.

### Format

```
with-relationships-clause ::=
    with-word                4.3
    relationships-word       4.3
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the with-meta-relationships IRD-schema-clause in Subsection 6.128.

## 6.33 Entity Selection Criteria Command-Clause

**Function** To specify the criteria by which entities are selected.

### Format

```
entity-selection-criteria-clause ::=
    select-word                4.3
    entity-selection
    [ where-word              4.3
      restriction-expression ]

entity-selection ::=
    all-entities-clause       6.42
    | entity-access-name-selection-clause 6.12
    | entity-descriptive-name-selection-clause 6.41
    | related-entities-clause 6.48

restriction-expression ::=
    restriction-expression
    boolean-operator          4.3
    restriction-expression
```

( restriction-expression )	
[ qualification-clause ]	
restriction-expression	
restriction-clause	
qualification-clause ::=	
entity-type-qualification-clause	6.1
relationship-type-qualification-clause	6.18
restriction-clause ::=	
entity-type-restriction-clause	6.34
relationship-existence-restriction-clause	6.35
attribute-restriction-clause	6.36
attribute-group-restriction-clause	6.37
text-attribute-substring-restriction-clause	6.40
order-restriction-clause	6.26
IRD-life-cycle-phase-restriction-clause	6.46
audit-attribute-restriction-clause	6.47
alternate-name-restriction-clause	6.49
quality-indicator-restriction-clause	6.31
irds-function-restriction-clause	6.77
entity-assigned-access-name-	
restriction-clause	6.78
entity-assigned-descriptive-name-	
restriction-clause	6.79
variation-name-restriction-clause	6.80
revision-number-restriction-clause	6.81

**Syntax Rules**

- (1) Parentheses shall be used to delimit compound expressions. Parentheses define hierarchical levels of control. At any control level, a qualification-clause is said to govern that control level and all levels under it. An entity-type restriction-clause can also govern value and/or substring restriction-clauses which are at lower control levels in the hierarchy.
- (2) An entity-type-qualification-clause shall not be required for an expression restricting values of attributes associated with entities if and only if:
  - (a) Value and/or substring restriction-clauses shall declared under the control of an entity-type restriction-clause, and
  - (b) All attribute-types or attribute group-types specified apply to all entity-types within the governing entity-type restriction-clause.
- (3) Parentheses shall be balanced.
- (4) Any attribute-type or attribute-group-type identified in a restriction-clause shall be associated with (all) the entity-type(s) or relationship-type(s) which govern the command-clause. The entity-type or relationship-type is said to qualify the attribute-type or attribute-group-type.
- (5) If an attribute (as opposed to a scan-pattern) is specified within a restriction-clause, it shall conform to the validation rules of the associated attribute-type.

- (6) An entity-type-qualification-clause shall not be governed by another entity-type-qualification-clause or a relationship-type-qualification-clause or an entity-type-restriction-clause.
- (7) A relationship-existence-restriction-clause or relationship-type-qualification-clause shall be governed by an entity-type-restriction-clause or entity-type-qualification-clause.
- (8) If a relationship-existence-restriction-clause or relationship-type-qualification-clause is specified, entity-type-1 of the relationship-type (or inverse) shall agree with the governing entity-type-restriction-clause or entity-type-qualification-clause.
- (9) The following restriction-clauses shall not be governed by relationship-type-qualification-clause:
  - (a) IRD-life-cycle-phase-restriction-clause
  - (b) Audit-attribute-restriction-clause
  - (c) Alternate-name-restriction-clause
  - (d) Quality-indicator-restriction-clause
  - (e) Entity-assigned-access-name-restriction-clause
  - (f) Entity-assigned-descriptive-name-restriction-clause
  - (g) Variation-name-restriction-clause
  - (h) Revision-number-restriction-clause
- (10) The order-restriction-clause shall be governed by a relationship-type qualification-clause.
- (11) An irds-function-restriction-clause which has an irds-function qualifier of entity-access-name-word or entity-descriptive-name-word shall not be governed by a relationship-type-qualification-clause.

#### **General Rules**

- (1) If the name of an entity (as opposed to a scan-pattern of an entity-access-name or entity-descriptive-name) is specified in entity-selection-clause, and the entity does not exist, an error condition shall occur.
- (2) If additional entity-access-names, entity-descriptive-names, and/or scan-patterns are specified after an entity name which is in error by General Rule (1) of this Subsection, processing shall continue on these entities and/or scan-patterns.

#### **Error and Warning Conditions**

- (1) **Error E01126:** Unrecognized or invalid IRD-schema-descriptor. An entity-type, relationship-type, attribute-type, attribute-group-type, or other IRD-schema-descriptor was expected. The designator did not identify a meta-entity.

- (2) Error E01174: Inconsistent IRD-schema-descriptors. A restriction-clause identified an IRD-schema-descriptor which was not associated with the governing IRD-schema-descriptor.
- (3) Error E01153: Inconsistent qualification. One of Syntax Rules (6) through (10) of this command-clause has been violated.
- (4) Error E01166: Nesting levels exceeded in expression. See Note (3).
- (5) Error E01133: Missing or unbalanced parentheses. Syntax Rule (3) of this command-clause has been violated.
- (6) Error E01167: Invalid use of irds function. Syntax Rule (11) has been violated.

## NOTES:

- (1) The entity-selection-clause shall be used to either select specific entities by name, browse entity-access-names or entity-descriptive-names, or by relationships to a specific entity.
- (2) Restriction-clauses then apply tests to the entities obtained via the entity-selection-clause to further narrow retrieval.
- (3) The implementor may restrict the number of nesting levels. The minimum shall be 5 levels of nesting.
- (4) This command-clause corresponds to the meta-entity-selection IRD-schema-clause in Subsection 6.118. Consult the notes for each of the entity-selection-clauses and restriction-clauses listed above to determine those which have corresponding IRD-schema-clauses.

### 6.34 Entity-Type Restriction Command-Clause

**Function** To restrict an entity-selection-clause to specified entity-types.

**Format**

```
entity-type-restriction-clause ::=
    entity-type-word                               4.3
    =
    entity-type-list
```

```
entity-type-list ::=
    entity-type-designator                         4.6
    [ , entity-type-list ]
```

**Syntax Rules**

- (1) Each entity-type-designator shall identify an ENTITY-TYPE meta-entity in the IRD Schema.

**General Rules**

- (1) An entity-type-restriction-clause which specifies more than one entity-types shall be equivalent to multiple entity-type-restriction-clauses, each specifying a single entity-type and connected by OR's.

- (2) Two entity-type-restriction-clauses shall not be connected by an AND.

**Error and Warning Conditions**

- (1) Error E01002: Invalid or unrecognized entity-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01040: Inconsistent command-clauses. General Rule (2) of this command-clause has been violated.

NOTE: This command-clause corresponds to the meta-entity-type-restriction IRD-schema-clause in Subsection 6.119.

**6.35 Relationship Existence Restriction Command-Clause**

**Function** To restrict selection of entities to those for which:

- (1) Relationships do exist.
- (2) No relationships exist.
- (3) Forward or inverse relationships exist.
- (4) Forward or inverse relationships do not exist.
- (5) Relationships of specified types exist.
- (6) Relationships of specified types do not exist.

**Format**

```
relationship-existence-restriction-clause ::=
    [ no-word ] 4.3
    [ forward-inverse-criterion ]
    relationships-word 4.3
    [ of-word 4.3
      [ relationship-type-word ] 4.3
      relationship-type-or-inverse ]
    exists-word 4.3

forward-inverse-criterion ::=
    forward-word 4.3
    | inverse-word 4.3

relationship-type-or-inverse ::=
    relationship-type-designator 4.6
    relationship-type-inverse-name 4.6
```

**Syntax Rules**

- (1) If forward-inverse-criterion is specified, then no relationship-type-designator or relationship-type-inverse-name shall be specified.



- (2) If forward-inverse-criterion and relationship-type-or-inverse are not specified, then "relationships of any type" shall be assumed.
- (3) Relationship-type-or-inverse shall be either the name of an existing RELATIONSHIP-TYPE meta-entity or the value of an INVERSE-NAME meta-attribute of a RELATIONSHIP-TYPE meta-entity in the IRD Schema.

### General Rules

- (1) If a relationship-type-designator or relationship-type-inverse-name is specified, this command-clause shall be governed by an entity-type-qualification-clause or an entity-type-restriction-clause. The governing entity-type shall be the first entity-type of the relationship-type.
- (2) If no-word is specified, the selection shall be restricted to those entities for which the specified relationships do not exist.
- (3) If no-word is not specified, the selection shall be restricted to those entities for which the specified relationships do exist.

### Error and Warning Conditions

- (1) Error E01040: Inconsistent command-clauses. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01105: Invalid relationship-type-designator or inverse. Syntax Rule (3) of this command-clause has been violated.
- (3) Error E01127: Improperly qualified command-clause. General Rule (1) of this command-clause has been violated.

NOTE: This command-clause corresponds to the meta-relationship-existence-restriction IRD-schema-clause in Subsection 6.122.

## 6.36 Attribute Restriction Command-Clause

**Function** To restrict the selection of entities and/or relationships based upon a test of attributes.

### Format

```

attribute-restriction-clause ::=
    attribute-type-designator           4.6
    relational-operator                 4.3
    [ only-word ]                       4.3
    test-value-list

test-value-list ::=
    test-value
    [ , test-value-list ]

test-value ::=
    numeric-literal                     4.2

```

short-string-literal	4.2
irds-name	4.2
null-mark	4.1
name-scan-mask	4.2
number-scan-mask	4.2
string-scan-mask	4.2

**Syntax Rules**

- (1) Attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema.
- (2) The ATTRIBUTE-TYPE identified by attribute-type-designator shall be associated either with the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE, or with an ATTRIBUTE-GROUP-TYPE which is associated with the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE. In this final case, the attribute-restriction-clause shall be a subclause of an attribute-group-restriction-clause, and the corresponding ATTRIBUTE-GROUP-TYPE shall be related in the IRD Schema to the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE.
- (3) The only relational operators which can be associated with name-scan-mask, number-scan-mask, string-scan-mask and null-mark shall be equals and not-equal.
- (4) If a scan-mask is specified, its format shall be consistent with the PICTURE and meta-attribute of the corresponding ATTRIBUTE-TYPE.
- (5) If an irds-name, numeric-literal or string-literal is specified, it shall specify a valid value for the attribute-type.
- (6) If the ATTRIBUTE-TYPE is NO within the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE, the only allowed relational operators shall be equals and not-equal.
- (7) Multiple test values shall be allowed only for an ATTRIBUTE-TYPE which is NO within the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE. In this case, the total number of test-values shall not exceed the value of the MAXIMUM-NUMBER-OF-OCCURRENCES meta-attribute within the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE.
- (8) If null-mark is specified, no other test-values shall be specified.
- (9) This command-clause can be used to test attributes within nonrepeating or repeating attribute-groups as long as attribute-type-designator is also not directly related to the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE in the IRD Schema.

**General Rules**

- (1) Null-mark shall be interpreted as a nonexistence indicator. Thus the command-clause "attribute-type-designator = null-mark" shall be interpreted as "attribute-type-designator does not exist". Conversely, the command-clause "attribute-type-designator < > null-mark" shall mean that "attribute-type-designator has been declared."
- (2) Only-word is only meaningful with a repeating attribute. If specified for an ATTRIBUTE-TYPE for which the value of SINGULAR is YES within the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE, then it shall be ignored.
- (3) If the ATTRIBUTE-TYPE identified by attribute-type-designator within the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE has the value NO for SINGULAR, then

- (a) The criterion, "attribute-type-designator = ONLY  $V_1, V_2, \dots, V_n$ ", shall be satisfied whenever attribute-type-designator exists with precisely the values  $V_1, V_2, \dots, V_n$ . If attribute-type-designator exists with any value not in the list  $V_1 \dots V_n$ , the criterion shall not be satisfied. Likewise, the criterion shall not be satisfied if attribute-type-designator exists with some but not all of the values in the list  $V_1 \dots V_n$ .
- (b) The criterion, "attribute-type-designator =  $V_1, V_2, \dots, V_n$ ", shall be satisfied whenever attribute-type-designator exists with at least the values  $V_1 \dots V_n$ .
- (c) The criterion, "attribute-type-designator  $\neq$  ONLY  $V_1, V_2, \dots, V_n$ ", shall be satisfied whenever attribute-type-designator exists with a set of values not equal to  $V_1 \dots V_n$ . Thus, if attribute-type-designator exists with any value not in the list  $V_1 \dots V_n$ , the criterion shall be satisfied. Likewise, the criterion shall be satisfied if attribute-type-designator exists with some but not all of the values in the list  $V_1 \dots V_n$ .
- (d) The criterion, "attribute-type-designator  $\neq V_1, V_2, \dots, V_n$ ", shall be satisfied whenever the attribute-type exists with none of the values  $V_1 \dots V_n$ . Thus if the attribute-type exists with one or more of the specified values, the criterion shall not be satisfied.

**Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01174: Inconsistent IRD-schema-descriptors. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01132: Too many values specified. Syntax Rule (7) of this command-clause has been violated.
- (4) Error E01168: Inconsistent test-value and relational operator. Syntax Rule (3) or (6) of this command-clause has been violated.
- (5) Error E01169: Invalid test-value. Syntax Rule (4), (5), or (8) of this command-clause has been violated.

NOTE: This command-clause corresponds to the meta-attribute-restriction IRD-schema-clause in Subsection 6.124.

**6.37 Attribute-Group Restriction Command-Clause**

**Function** To restrict the selection of entities and/or relationships based upon a test of attributes within an attribute-group.

**Format**

```

attribute-group-restriction-clause ::=
    for-word                               4.3
    attribute-group-type-designator         4.6
    component-attribute-restriction-expression
    
```

```

component-attribute-restriction-expression ::=
    attribute-restriction-clause                6.36
    | ( component-attribute-restriction-expression )
    | component-attribute-restriction-expression
      boolean-operator                          4.3
      component-attribute-restriction-expression

```

### Syntax Rules

- (1) Attribute-group-type-designator shall be the name of an ATTRIBUTE-GROUP-TYPE meta-entity in the IRD Schema.
- (2) The ATTRIBUTE-GROUP-TYPE meta-entity identified by attribute-group-type-designator shall be related in the IRD Schema to the qualifying ENTITY-TYPE or RELATIONSHIP-TYPE.
- (3) The expression shall be terminated by either the command-terminator or a right parentheses which is not balanced by a left parentheses which occurs after attribute-group-type-designator. If any other restriction-clauses are specified after an attribute-group-restriction-clause, then the entire attribute-group-restriction-clause shall be enclosed in parentheses.
- (4) Each attribute-type-designator specified in an attribute-restriction-clause within this command-clause shall identify an ATTRIBUTE-TYPE meta-entity which is related in the IRD Schema to the ATTRIBUTE-GROUP-TYPE meta-entity identified by attribute-group-type-designator.

### General Rules

- (1) For a repeating attribute-group, component-attribute-restriction-expressions connected by and-word shall be assumed to refer to the same occurrence of the attribute-group.

/\* For example, suppose a repeating ATTRIBUTE-GROUP-TYPE named AG has component ATTRIBUTE-TYPEs X and Y. Then the restriction-clause:

$$(\text{FOR } AG \ X = x1 \ \text{AND } Y = y1)$$

is fulfilled by the repeating attribute-group:

$$AG = (x0,y0)(x1,y1)(x2,y2)$$

but not by the repeating attribute-group:

$$AG = (x0,y1)(x1,y2)(x2,y0).$$

In order to have both of the above repeating attribute-groups fulfill the restriction, the command-clause would have to be:

(FOR AG (X = x1 AND Y = y1) OR ((X = x1 AND Y <> y1) AND (X <> x1 AND Y = y1)))  
\*/

**Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01174: Inconsistent IRD-schema-descriptors. Syntax Rule (2) or (4) of this command-clause has been violated.
- (3) Error E01133: Missing or unbalanced parentheses.

NOTE: This command-clause corresponds to the meta-attribute-group-restriction IRD-schema-clause in Subsection 6.123.

**6.38 Using IRD-VIEWS Command-Clause**

**Function** To identify those IRD-VIEWS which are to be enabled within a command.

**Format**

```

using-IRD-views-clause ::=
    using-word                4.3
    IRD-view-word            4.3
    =
    IRD-view-option

IRD-view-option ::=
    all-word                4.3
    | IRD-view-name-list

IRD-view-name-list ::=
    IRD-view-name
    [ , IRD-view-name-list ]

IRD-view-name ::= assigned-access-name 4.4

```

**Syntax Rules** None.

**General Rules**

- (1) Each IRD-view-name specified shall identify an existing IRD-VIEW entity in the IRD.
- (2) Each specified IRD-view-name shall identify an IRD-VIEW which is related to the IRDS-USER entity by which the user executing the command has identified himself. An IRD-VIEW which does not fulfill this criterion shall be treated as nonexistent.
- (3) Each specified IRD-VIEW shall have a PARTITION-NAME attribute equal to the name of an IRD-Partition defined in the IRD Schema.
- (4) If all-word is specified, then all IRD-VIEW entities related to the IRDS-USER by which the user is identified shall be used. This shall be equivalent to specifying each of these IRD-VIEWS individually.

### Error and Warning Conditions

- (1) Error E01170: IRD-View does not exist. See General Rules (1) and (2) of this command-clause.
- (2) Error E01125: IRD-View does not identify a valid IRD-Partition. General Rule (3) has been violated.

NOTE: This command-clause corresponds to the using-IRD-schema-views IRD-schema-clause in Subsection 6.132.

### 6.40 Text Attribute Substring Restriction Command-Clause

**Function** To use a substring criterion on a text attribute-type to select entities and/or relationships.

#### Format

```
text-attribute-substring-restriction-clause ::=
    attribute-type-designator           4.6
    [ line-range-clause ]              6.21
    equals-or-not-equal
    string-scan-mask                   4.2

equals-or-not-equal ::=
    equals                               4.3
    | not-equal                          4.3
```

#### Syntax Rules

- (1) Attribute-type-designator shall be the name of an ATTRIBUTE-TYPE meta-entity in the IRD Schema.
- (2) The ATTRIBUTE-TYPE meta-entity identified by attribute-type-designator shall have a PICTURE meta-attribute of TEXT.

#### General Rules

- (1) If no line-range-clause is specified, the range:

"(FIRST THROUGH LAST)"

shall be assumed.

### Error and Warning Conditions

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. Syntax Rule (1) for this command-clause has been violated.
- (2) Error E01176: Inconsistent subclause. Syntax Rule (2) for this command-clause has been violated. (A line range subclause was specified for a non-text-attribute.)

**6.41 Entity-Descriptive-Name Selection Command-Clause**

**Function** To retrieve entities by entity-descriptive-name or by a substring within the entity-descriptive-name.

**Format**

```

entity-descriptive-name-selection-clause ::=
    entities-word                4.3
    [ with-word ]                4.3
    descriptive-name-word        4.3
    =
    descriptive-name-scan-pattern-list

descriptive-name-scan-pattern-list ::=
    descriptive-name-pattern
    [ , descriptive-name-scan-pattern-list ]

descriptive-name-pattern ::=
    name-scan-pattern            4.4
    | entity-descriptive-name    4.4

```

**Syntax Rules**

- (1) Entity-descriptive-names and name-scan-patterns can be mixed in the list.
- (2) Each name-scan-mask in each name-scan-pattern shall not exceed the maximum number of characters which is allowed for the corresponding part of an entity-descriptive-name.
- (3) If no substitution character is given in the variation-name part of the name-scan-mask, then it shall specify a valid variation-name.

**General Rules**

- (1) If no substitution character is specified, the name-scan-pattern shall be interpreted as the entity-descriptive-name of an entity which exists in the IRD.

**Error and Warning Conditions**

- (1) Error E01005: Invalid name format. One or more of the following situations has occurred:
  - (a) The entity-assigned-descriptive-name part contained an invalid character.
  - (b) The variation-name part contained an invalid character.
  - (c) Parentheses were not balanced in the version-identifier.
  - (d) The colon was missing in the version-identifier.
  - (e) The length of the entity-assigned-descriptive-name or variation-name exceeded the maximum allowed.

- (f) The revision-number part of the name did not contain an unsigned integer or an asterisk.
- (2) Error E01009: Entity does not exist or is not visible. No substitution-character was specified and the specified entity-descriptive-name did not exist or was not visible.
- (3) Error E01158: Variation-name does not exist. No substitution-character was specified in the variation-name part of the entity-descriptive-name or name-scan-pattern. The variation-name does not exist.

### 6.42 All Entities Command-Clause

**Function** To identify that all entities within the effective IRD-VIEW or specified set of IRD-VIEWS are to be selected.

**Format**

```
all-entities-clause ::=
    all-word                4.3
    [ entities-word ]      4.3
```

**Syntax Rules** None.

**General Rules**

- (1) If no using-IRD-views-clause is specified as part of the command, only those entities within the effective IRD-VIEW shall be selected. If a using-IRD-views-clause is specified within the command, then any entity which is within one or more of the specified IRD-VIEWS shall be selected.

**Error and Warning Conditions** None.

### 6.43 New Relationship Attributes Command-Clause

**Function** To identify those attributes and/or attribute-groups which are to be created as part of an add-relationship-command.

**Format**

```
new-relationship-attributes-clause ::=
    with-word                4.3
    [attributes-word]        4.3
    new-attribute-or-attribute-group-list

new-attribute-or-attribute-group-list ::=
    new-attribute-or-attribute-group
    [ , new-attribute-or-attribute-group-list ]

new-attribute-or-attribute-group ::=
    simple-attribute-clause  6.3
    | new-repeating-attribute-clause  6.4
    | new-simple-attribute-group-clause  6.5
```



new-repeating-attribute-group-clause	6.6
new-text-attribute-clause	6.7

**Syntax Rules**

- (1) The sequence attribute or order-clause shall not be specified here.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01044: Invalid, missing, or misplaced command-clause. With-word expected and not recognized or attributes-word misspelled.

**6.44 Modified Relationship Attributes Command-Clause**

**Function** To identify those attributes and/or attribute-groups of a relationship which are to be created, modified, or deleted by a modify-relationship-command.

**Format**

modified-relationship-attributes-clause ::=	
with-word	4.3
[ attributes-word ]	4.3
modified-attributes-and-or-order-clause	
modified-attributes-and-or-order-clause ::=	
new-sequence-attribute-or-order-clause	
modified-attributes	
new-sequence-attribute-or-order-clause	
modified-attributes	
new-sequence-attribute-or-order-clause ::=	
sequence-attribute	
order-clause	6.17
sequence-attribute ::=	
simple-attribute-clause	6.3
modified-attributes ::=	
modified-attribute-or-attribute-group-list	
modified-attribute-or-attribute-group-list ::=	
modified-attribute-or-attribute-group	
[ , modified-attribute-or-attribute-group-list ]	
modified-attribute-or-attribute-group ::=	
simple-attribute-clause	6.3
modified-repeating-attribute-clause	6.8
modified-simple-attribute-group-clause	6.9
modified-repeating-attribute-group-clause	6.10
modified-text-attribute-clause	6.11
text-attribute-resequence-clause	6.29

**Syntax Rules**

- (1) If specified, the sequence-attribute-or-order-clause shall be specified prior to any other relationship attributes.

**General Rules**

- (1) The value of the new sequence-attribute or the new value specified in the order-clause shall not be the same as in any other relationship of the type being modified with the same first entity.

**Error and Warning Conditions**

- (1) Error E01044: Invalid, missing, or misplaced command-clause. With-word expected and not recognized or attributes-word misspelled.
- (2) Error E01177: Invalid order value. Violation of General Rule (1).

**6.45 Relationship Selection Command-Clause**

**Function** To select relationships.

**Format**

```

relationship-selection-clause ::=
    select-word                               4.3
    [ all-word                                4.3
      [ forward-or-inverse-word ]
    ]
    relationships-word                         4.3
    for-word                                  4.3
    specified-entities
    [ where-word                              4.3
      relationship-restriction-expression
    ]

forward-or-inverse-word ::=
    forward-word                               4.3
    | inverse-word                             4.3

specified-entities ::=
    name-scan-pattern-list

name-scan-pattern-list ::=
    name-pattern
    [ , name-scan-pattern-list ]

name-pattern ::=
    entity-access-name                         4.4
    | name-scan-pattern                       4.4

relationship-restriction-expression ::=
    relationship-restriction-clause
    
```

( relationship-restriction-expression )	
relationship-restriction-expression	
boolean-operator	4.3
relationship-restriction-expression	
relationship-restriction-clause ::=	
relationship-type-restriction-clause	6.50
attribute-restriction-clause	6.36
attribute-group-restriction-clause	6.37
order-restriction-clause	6.26
text-attribute-substring-restriction-clause	6.40
irds-function-restriction-clause	6.77

### Syntax Rules

- (1) If all-word is specified, then relationship-restriction-expression shall not be specified.
- (2) Each attribute-restriction-clause, attribute-group-restriction-clause, order-restriction-clause, and text-attribute-substring-restriction-clause shall be part of a relationship-restriction-expression which is connected by and-word to a relationship-type-restriction-clause. Thus a relationship-type-restriction-clause shall govern all other kinds of relationship-restriction-clauses.
- (3) Two relationship-restriction-expressions governed by different relationship-type-restriction-clauses shall be connected by or-word. (A relationship cannot have more than one relationship-type.)
- (4) An order-restriction-clause shall be specified only if it is governed by a relationship-type-clause which identifies a sequenced RELATIONSHIP-TYPE or a RELATIONSHIP-CLASS-TYPE associated with a sequenced RELATIONSHIP-TYPE which does not have a special sequencing ATTRIBUTE-TYPE.
- (5) Each ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE specified in a relationship-restriction-clause shall be associated with a RELATIONSHIP-TYPE which is specified or associated with the RELATIONSHIP-CLASS-TYPE specified in the governing relationship-type-restriction-clause.

### General Rules

- (1) If the name of an entity (as opposed to a scan-pattern of an entity-access-name) is specified, and the entity does not exist, an error condition shall be raised.
- (2) If additional entity-access-names and/or scan patterns are specified after an entity-access-name which is in error by General Rule (1) of this Command-Clause, processing shall continue on these entity-access-names and/or scan-patterns.
- (3) If irds-function-restriction-clause is used, the irds-function qualifier shall be a specific attribute-type-designator or attribute-group-type-designator.

### Error and Warning Conditions

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. A relationship-type, relationship-class-type, attribute-type, attribute-group-type, or ORDER was expected. The designator did not identify a valid meta-entity for this command-clause.

- (2) Error E01174: Inconsistent IRD-schema-descriptors. One of Syntax Rules (2) through (6) for this command-clause was violated.
- (3) Error E01133: Missing or unbalanced parentheses.
- (4) Error E01142: Inconsistent command-clause format. Syntax Rule (1) for this command-clause has been violated.
- (5) Error E01009: Entity does not exist or is not visible. General Rule (1) for this command-clause has been violated.
- (6) Error E01167: Invalid use of irds function. General Rule (3) for this command-clause has been violated.

### 6.46 IRD Life-Cycle-Phase Restriction Command-Clause

**Function** To select entities based upon which IRD-life-cycle-phase they are in.

**Format**

```
IRD-life-cycle-phase-restriction-clause ::=
    IRD-word                               4.3
    life-cycle-phase-word                  4.3
    relational-operator                    4.3
    IRD-life-cycle-phase-designator        4.6
```

**Syntax Rules**

- (1) IRD-life-cycle-phase-designator shall be defined in the IRD Schema.

**General Rules**

- (1) Each UNCONTROLLED IRD-life-cycle-phase shall be defined to be less-than the CONTROLLED IRD-life-cycle-phase. The CONTROLLED IRD-life-cycle-phase shall also be less-than the ARCHIVED IRD-life-cycle-phase.

**Error and Warning Conditions**

- (1) Error E01033: Unrecognized IRD life-cycle-phase-designator. Syntax Rule (1) for this command-clause has been violated.

NOTE: This command-clause corresponds to the IRD-schema-life-cycle-phase-restriction IRD-schema-clause in Subsection 6.129.

### 6.47 Audit Attribute Restriction Command-Clause

**Function** To restrict the selection of entities and/or relationships based upon a test of values of audit-attributes.

**Format**

```
audit-attribute-restriction-clause ::=
```

audit-attribute-type	
relational-operator	4.3
test-value	
audit-attribute-type ::=	
attribute-type-designator	4.6
test-value ::=	
numeric-literal	4.2
short-string-literal	4.2
irds-name	4.2
null-mark	4.1
name-scan-mask	4.2
number-scan-mask	4.2
string-scan-mask	4.2

### Syntax Rules

- (1) Attribute-type-designator shall identify an ATTRIBUTE-TYPE in the IRD Schema which is either a standard audit attribute-type or an implementation-defined audit attribute-type.
- (2) The ATTRIBUTE-TYPE identified by attribute-type-designator shall be associated with the qualifying ENTITY-TYPE.
- (3) The only relational operators which shall be associated with name-scan-mask, number-scan-mask, string-scan-mask and null-mark shall be equals and not-equal.
- (4) If a scan-mask is specified, its format shall be consistent with the PICTURE and meta-attribute of the corresponding ATTRIBUTE-TYPE.
- (5) If an irds-name, numeric-literal or short-string-literal is specified, it shall specify a valid value for the attribute-type.

### General Rules

- (1) Audit attributes which have no value shall be treated as if they were nonexistent in the entity. For example, an entity which has not been modified shall not have a LAST-MODIFIED-BY audit attribute.
- (2) Null-mark shall be interpreted as a nonexistence indicator. Thus, the command-clause "attribute-type-designator = null-mark" shall be interpreted as "attribute-type-designator does not exist". Conversely, the command-clause "attribute-type-designator < > null-mark" shall mean that "attribute-type-designator has been declared."

### Error and Warning Conditions

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. Syntax Rule (1) for this command-clause has been violated.
- (2) Error E01174: Inconsistent IRD-schema-descriptors. Syntax Rule (2) for this command-clause has been violated.
- (3) Error E01132: Too many values specified. A list of test-values has been specified.

- (4) Error E01168: Inconsistent test-value and relational operator. Syntax Rule (3) for this command-clause has been violated.
- (5) Error E01169: Invalid test-value. Syntax Rule (4) or (5) for this command-clause has been violated.

### 6.48 Related Entities Command-Clause

**Function** To specify retrieval of entities related to a specified entity, and how those entities are related to the specified entity.

#### Format

```

related-entities-clause ::=
    entities-word                4.3
    [ directly-word ]           4.3
    related-word                4.3
    to-word
    entity-access-names-or-list
    [ via-word                   4.3
      path-list ]

entity-access-names-or-list ::=
    entity-access-name-list
    | using-list-clause         6.13

entity-access-name-list ::=
    entity-access-name          4.4
    [ , entity-access-name-list ]

path-list ::=
    path
    [ , path-list ]

path ::=
    relationship-type-designator 4.6
    | relationship-type-inverse-name 4.6
    | relationship-class-type-designator 4.6
    | relationship-class-type-inverse-name 4.6
    
```

#### Syntax Rules

- (1) Each path shall be either the name of a RELATIONSHIP-TYPE or RELATIONSHIP-CLASS-TYPE meta-entity in the IRD Schema, or the value of an INVERSE-NAME meta-attribute of a RELATIONSHIP-TYPE or RELATIONSHIP-CLASS-TYPE meta-entity.
- (2) If directly-word is specified, no path specification shall be required.

#### General Rules

For all General Rules of this Command-Clause the following notation is used:

*E(s)* is an entity specified in the command-clause or in the specified list.

$E(i)$ ,  $E(j)$  are other entities.

$R$ ,  $R(j)$ ,  $R(k)$  are (forward) relationship-types.

$E(i) R E(j)$  denotes a relationship of relationship-type  $R$  in which  $E(i)$  is entity-1 and  $E(j)$  is entity-2.

$R'$  shall denote the inverse of relationship-type  $R$ , i.e.,  $E(i) R' E(j)$ , if and only if  $E(j) R E(i)$ .

$P(1) \dots P(n)$  denotes the list of paths.

(1) If directly-word is specified, then:

- (a) If no paths are specified, then each  $E(i)$  shall be selected for which there are relationships of some type  $R$  such that the relationship " $E(i) R E(s)$ " or " $E(s) R E(i)$ " exists.
- (b) If paths  $P(j)$ ,  $1 \leq j \leq m$ , are specified, then each  $E(i)$  shall be selected for which the relationship " $E(i) P(j) E(s)$ " exists. Note that if  $P(j)$  is an inverse of a relationship-type  $R$  or a relationship-class-type  $C$ , this is equivalent to the condition that the relationship " $E(s) R E(i)$ " or " $E(s) R E(i)$ " exist.

(2) If directly-word is not specified, then:

(a) If no paths are specified, then all related entities shall be selected in the following manner:

- (i)  $E(s)$  is exploded. That is, given  $E(s)$ , all  $E(i)$  shall be selected such that there is a sequence of forward relationships which connect  $E(i)$  to  $E(s)$  as follows:

$$\begin{array}{l} E(s) \quad R(1) \quad E(1) \\ E(1) \quad R(2) \quad E(2) \\ \dots \\ E(i-1) \quad R(n) \quad E(i). \end{array}$$

- (ii)  $E(s)$  is imploded. That is, given  $E(s)$ , all  $E(i)$  shall be selected such that there is a sequence of forward relationships which connect  $E(i)$  to  $E(s)$  as follows:

$$\begin{array}{l} E(i) \quad R(1) \quad E(1) \\ E(1) \quad R(2) \quad E(2) \\ \dots \\ E(i-1) \quad R(n) \quad E(s). \end{array}$$

- (iii) Either the explosion or the implosion may be performed first.
- (iv) The results of the explosion and the implosion shall be pruned. That is, if an entity has already been selected in an explosion or implosion, it shall not be selected again.

(b) If paths  $P(1) \dots P(n)$  are specified, then entities shall be selected by imploding along the specified paths. That is, all entities  $E(i)$  shall be selected for which there exists relationships:

$$\begin{array}{c}
 E(i) \quad P(i1) \quad E(i1) \\
 E(i1) \quad P(i2) \quad E(i2) \\
 \dots \\
 E(im) \quad P(im+1) \quad E(s)
 \end{array}$$

where each  $P(ij)$  is one of  $P(1) \dots P(n)$ . Note that the same paths may be repeated in any such sequence of relationships. Thus, "recursive" relationship-types are traversed. This implosion process shall also be pruned.

- (3) If  $E(s)$  is of ENTITY-TYPE  $T$ , and a path list is specified which does not specify or (in the case of a RELATIONSHIP-CLASS-TYPE or an inverse of a RELATIONSHIP-CLASS-TYPE) includes a RELATIONSHIP-TYPE or inverse of a RELATIONSHIP-CLASS-TYPE which is associated with  $T$  in the IRD Schema, then an execution-time error shall occur. If  $E(s)$  occurs in a list, processing shall continue with the next entity in the list.
- (4) Each  $E(s)$  shall exist. If any given  $E(s)$  does not exist, an error condition shall be raised on that entity, and processing shall continue with the next  $E(s)$ .

**Error and Warning Conditions**

- (1) Error E01178: Invalid path specified. Syntax Rule (1) for this command-clause has been violated.
- (2) Error E01120: Entity not connectable via specified path(s). General Rule (3) for this command-clause has been violated.
- (3) Error E01009: Entity does not exist or is not visible. General Rule (4) for this command-clause has been violated.

NOTE: The selection of indirectly-related entities may be performed according to an implementor-defined strategy. For example, either a depth-first or breadth-first strategy may be employed.

**6.49 Alternate-Name Restriction Command-Clause**

**Function** To provide an easy method to test the attributes of the ALTERNATE-NAME attribute within the IDENTIFICATION-NAMES attribute-group-type.

**Format**

```

alternate-name-restriction-clause ::=
    alternate-name-word                               4.3
    =
    alternate-name
    [ in-word                                         4.3
      context-word                                     4.3
      context ]

alternate-name ::=
    irds-name                                         4.2
    | name-scan-mask                                  4.2

context ::=
    irds-name                                         4.2
    
```



short-string-literal	4.2
string-scan-mask	4.2

### Syntax Rules

- (1) Alternate-name is a test value for the ALTERNATE-NAME attribute-type within the IDENTIFICATION-NAMES attribute-group-type.
- (2) Context is a test value for the ALTERNATE-NAME-CONTEXT attribute-type within the IDENTIFICATION-NAMES attribute-group-type.
- (3) Context shall be a valid value for the ALTERNATE-NAME-CONTEXT attribute-type.

### General Rules

- (1) Unlike attribute-restriction-clauses and attribute-group-restriction-clauses, this command-clause need not be governed by an entity-type-qualification-clause or relationship-type-qualification-clause in order to establish the appropriate qualification.
- (2) If it is governed by an entity-type-qualification-clause, the restriction shall apply only to the specified entity-type(s). All relationship-types with the specified entity-type(s) as entity-type-2 shall be implicitly identified.
- (3) If it is governed by a relationship-type, the IDENTIFICATION-NAMES attribute-group-type shall be related in the IRD Schema to the corresponding RELATIONSHIP-TYPE meta-entity, and the test shall be restricted to only that relationship-type.

### Error and Warning Conditions

- (1) Error E01128: Invalid attribute or format. Either alternate-name did not contain a valid name format or context did not contain a valid value for the ALTERNATE-NAME-CONTEXT attribute-type.

## 6.50 Relationship-Type Restriction Command-Clause

**Function** To restrict the selection of relationships to a particular relationship-type and to govern subsequent restriction-clauses.

### Format

```

relationship-type-restriction-clause ::=
    relationship-type-word                4.3
    is-type-or-in-class-specification

is-type-or-in-class-specification ::=
    is-type-specification
    | is-in-class-specification

is-type-specification ::=
    =
    relationship-type-or-inverse

```

relationship-type-or-inverse ::=	
relationship-type-designator	4.6
relationship-type-inverse-name	4.6
is-in-class-specification ::=	
in-word	4.3
relationship-class-type-or-inverse	
relationship-class-type-or-inverse ::=	
relationship-class-type-designator	4.6
relationship-class-type-inverse-name	4.6

### Syntax Rules

- (1) Relationship-type-designator shall be the name of a RELATIONSHIP-TYPE meta-entity in the IRD Schema.
- (2) Relationship-type-inverse-name shall be the value of the INVERSE-NAME meta-attribute of a RELATIONSHIP-TYPE meta-entity in the IRD Schema.
- (3) Relationship-class-type-designator shall be the name of a RELATIONSHIP-CLASS-TYPE meta-entity in the IRD Schema.
- (4) Relationship-class-type-inverse-name shall be the value of the INVERSE-NAME meta-attribute of a RELATIONSHIP-CLASS-TYPE meta-entity in the IRD Schema.

### General Rules

- (1) If any attribute-type-designator or attribute-group-type-designator is specified in a restriction-clause governed by this command-clause, then the ATTRIBUTE-TYPE or ATTRIBUTE-GROUP-TYPE meta-entity shall be:
  - (a) Associated with the corresponding RELATIONSHIP-TYPE meta-entity if relationship-type-designator is specified.
  - (b) Associated with the corresponding RELATIONSHIP-TYPE meta-entity with the specified INVERSE-NAME meta-attribute if relationship-type-inverse-name is specified.
  - (c) Associated with all RELATIONSHIP-TYPE meta-entities which are associated with the corresponding RELATIONSHIP-CLASS-TYPE if relationship-class-type-designator is specified.
  - (d) Associated with all RELATIONSHIP-TYPE meta-entities which are associated with the corresponding RELATIONSHIP-CLASS-TYPE with the specified INVERSE-NAME meta-attribute if relationship-class-type-designator is specified.
- (2) If an order-restriction-clause is governed by this command-clause, then:
  - (a) If relationship-type-designator is specified, the corresponding RELATIONSHIP-TYPE meta-entity shall be sequenced without using a special sequencing ATTRIBUTE-TYPE.

- (b) If relationship-type-inverse-name is specified, the corresponding RELATIONSHIP-TYPE meta-entity with an INVERSE-NAME meta-attribute of the specified value shall be sequenced without using a special sequencing ATTRIBUTE-TYPE.
- (c) If relationship-type-designator is specified, each RELATIONSHIP-TYPE meta-entity associated with the identified RELATIONSHIP-CLASS-TYPE meta-entity shall be sequenced without using a special sequencing ATTRIBUTE-TYPE.
- (d) If relationship-class-type-inverse-name is specified, each RELATIONSHIP-TYPE meta-entity associated with the RELATIONSHIP-CLASS-TYPE meta-entity with an INVERSE-NAME meta-attribute of the specified value shall be sequenced without using a special sequencing ATTRIBUTE-TYPE.

**Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD Schema description. Syntax Rules (1) through (4) for this command clause have been violated.
- (2) Error E01127: Improperly qualified command-clause. General Rule (1)(a), (1)(b), (2)(a), or (2)(b) for this command-clause has been violated. See Note (1).
- (3) Error E01180: Command-Clause(s) not consistent for all relationship-types in relationship-class-type. General Rule (1)(c), (1)(d), (2)(c), or (2)(d) for this command-clause has been violated. See Note (2).

**NOTES:**

- (1) Violation of General Rules (1)(a), (1)(b), (2)(a), and (2)(b) shall be syntax level violations.
- (2) Violation of conditions (c) or (d) of General Rules (1) and/or (2) shall be treated as an execution error. It will be sensed only if a relationship was encountered where the relationship-type violated one or both of these rules. The command-clause shall be ignored whenever a relationship of a relationship-type which violates this rule is encountered. The corresponding message shall be produced only once for each entity being processed by the relationship-selection-clause where the violation occurs.

**6.51 In IRD Command-Clause \*\*EX/IM\*\***

**Function** To identify the IRD whose IRD Schema is to be used in a create-IRD-command or in a check-IRD-schema-compatibility-command.

**Format**

```

in-IRD-clause ::=
    in-word           4.3
    IRD-word          4.3
    IRD-name
    [ location-clause ] 6.57

IRD-name ::=
    /* implementor-defined-format */
    
```

### Syntax Rules

- (1) IRD-name shall conform to implementor-defined rules for naming an IRD.
- (2) The implementor may require location-clause.

### General Rules

- (1) The identified IRD shall exist.
- (2) If location-clause is specified, the identified IRD shall exist at the specified location.

### Error and Warning Conditions

- (1) Error E01039: Invalid IRD name. Syntax Rule (1) for this command-clause has been violated.
- (2) Error E01044: Invalid, missing, or misplaced command-clause. See Syntax Rule (2).
- (3) Error E01181: Identified IRD does not exist at specified or assumed location. See General Rules (1) and (2).

## 6.52 In File Command-Clause **\*\*EX/IM\*\***

**Function** To identify the IRD-schema-export-file which is to be used in a create-IRD-command or in a check-IRD-schema-compatibility-command.

### Format

```
in-file-clause ::=
    in-word           4.3
    file-word        4.3
    file-name
    [ location-clause ] 6.57

file-name ::=
    /* implementor-defined-format */
```

### Syntax Rules

- (1) File-name shall conform to implementor-defined rules for naming data files.
- (2) The implementor may require location-clause.

### General Rules

- (1) The specified file shall exist at the specified or assumed location.
- (2) The specified file shall be an IRD-schema-export-file.

### Error and Warning Conditions

- (1) Error E01182: Invalid file-name. See Syntax Rule (1) for this command-clause.

- (2) Error E01044: Invalid, missing, or misplaced command-clause. See Syntax Rule (2) for this command-clause.
- (3) Error E01183: Specified file does not exist at specified or assumed location. General Rule (1) for this command-clause has been violated.
- (4) Error E01184: Invalid file format. General Rule (2) for this command-clause has been violated.

### 6.53 IRD Schema Source Command-Clause

**Function** To identify the source of the IRD Schema to be used in a create-IRD-command.

**Format**

```

IRD-schema-source-clause ::=
    IRD-schema-word                4.3
    [ is-word ]                    4.3
    IRD-schema-source-option

IRD-schema-source-option ::=
    in-file-clause **EX/IM**      6.52
    | in-IRD-clause **EX/IM**    6.51
    | minimal-word                4.3

```

**Syntax Rules** None.

**General Rules**

- (1) If minimal-word is specified in IRD-schema-source-option, the IRD shall be created with the Minimal IRD Schema.

**Error and Warning Conditions** None.

### 6.54 Load IRD Command-Clause \*\*EX/IM\*\*

**Function** To identify that a newly created IRD is to be loaded with the contents of IRD-export-file.

**Format**

```

load-IRD-clause ::=
    load-word                4.3
    [ IRD-word ]            4.3
    [ from-word ]          4.3
    file-word                4.3
    file-name
    [ location-clause ]    6.57

file-name ::=
    /* implementor-defined-format */

```

### Syntax Rules

- (1) File-name shall conform to implementor-defined rules for data file names.
- (2) The implementor may require location-clause to be specified.

### General Rules

- (1) The specified file shall exist at the specified or assumed location.
- (2) The specified file shall be an IRD-export-file.

### Error and Warning Conditions

- (1) Error E01182: Invalid file-name. See Syntax Rule (1) for this command-clause.
- (2) Error E01044: Invalid, missing, or misplaced command-clause. See Syntax Rule (2) for this command-clause.
- (3) Error E01183: Specified file does not exist at specified or assumed location. General Rule (1) for this command-clause has been violated.
- (4) Error E01184: Invalid file format. General rule (2) for this command-clause has been violated.

## 6.55 IRD Schema Export File Command-Clause **\*\*EX/IM\*\***

**Function** To specify an IRD-schema-export-file-name.

### Format

```
IRD-schema-export-file-clause ::=
    IRD-schema-word           4.3
    export-word               4.3
    file-word                 4.3
    =
    IRD-schema-export-file-name
    [ location-clause ]      6.57

IRD-schema-export-file-name ::=
    /* implementor-defined-format */
```

### Syntax Rules

- (1) The IRD-schema-export-file-name shall conform to implementor-defined rules for naming data files.
- (2) The implementor may require location-clause.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01185: Invalid IRD-schema-export-file-name. Syntax Rule (1) has been violated.
- (2) Error E01044: Invalid, missing, or misplaced command-clause. See Syntax Rule (2) of this command-clause.

**6.56 IRD Export File Command-Clause \*\*EX/IM\*\***

**Function** To specify an IRD-export-file-name.

**Format**

```

IRD-export-file-clause ::=
    IRD-word                4.3
    export-word             4.3
    file-word               4.3
    =
    IRD-export-file-name
    [ location-clause ]    6.57

IRD-export-file-name ::=
    /* implementor-defined-format */

```

**Syntax Rules**

- (1) The IRD-export-file-name shall conform to implementor-defined rules for naming data files.
- (2) The implementor may require location-clause.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01186: Invalid IRD-export-file-name. Syntax Rule (1) has been violated.
- (2) Error E01044: Invalid, missing, or misplaced command-clause. See Syntax Rule (2) of this command-clause.

**6.57 Location Command-Clause \*\*EX/IM\*\***

**Function** To locate a specified file or IRD if the file-name or IRD-name is not sufficient.

**Format**

```

location-clause ::=
    /* implementor-defined-format */

```

**Syntax Rules** Implementor-defined.

**General Rules** Implementor-defined.

**Error and Warning Conditions**    Implementor defined.

**6.58 File Title Suffix Command-Clause   \*\*EX/IM\*\***

**Function**    To specify an optional suffix for the File Title found in the IRD-schema-export-file and the IRD-export-file.

**Format**

```

file-title-suffix-clause
    suffix-word                4.3
    =
    short-string-literal      4.2
    
```

**Syntax Rules**    None.

**General Rules**

- (1) The string-literal specified shall be appended as a suffix on the File Title in both the IRD-schema-export-file and the IRD-export-file.
- (2) If only the null-string ("") is specified, the system shall regard this command-clause as not having been entered.

**Error and Warning Conditions**    None.

**6.59 Other IRD Schema Command-Clause   \*\*EX/IM\*\***

**Function**    To identify where the other IRD Schema to be used in a check-IRD-schema-compatibility-command resides, and whether it is to be used as a source-IRD Schema or target-IRD Schema in the compatibility check.

**Format**

```

other-IRD-schema-clause ::=
    [ source-or-target-option ]
    IRD-schema-word                4.3
    [ is-word ]                    4.3
    other-IRD-schema-location

source-or-target-option ::=
    source-word                    4.3
    | target-word                  4.3

other-IRD-schema-location ::=
    in-IRD-clause                  6.51
    | in-file-clause              6.52
    | minimal-word                4.3
    
```

**Syntax Rules**

- (1) If source-or-target option is not specified, source-word shall be assumed.



**General Rules**

- (1) If source-word is specified or assumed, the IRD Schema specified by other-IRD-schema-location shall be used as the source-IRD Schema in the compatibility check.
- (2) If target-word is specified or assumed, the IRD Schema specified by other-IRD-schema-location shall be used as the target-IRD Schema in the compatibility check.
- (3) If minimal-word is specified in other-IRD-schema-location, the Minimal IRD Schema shall be used as the other IRD Schema in the compatibility check.

**Error and Warning Conditions** None.

**6.60 Exclude Relationships of Type Command-Clause**

**Function** To identify the types of relationships which shall be excluded in an extraction of IRD descriptors.

**Format**

```

exclude-relationships-of-type-clause ::=
    exclude-word                               4.3
    [ relationships-word                       4.3
      of-word ]                               4.3
    relationship-type-or-class-type-list

relationship-type-or-class-type-list ::=
    relationship-type-or-class-type
    [ , relationship-type-or-class-type-list ]

relationship-type-or-class-type ::=
    relationship-type-designator               4.6
    | relationship-class-type-designator      4.6

```

**Syntax Rules** None.

**General Rules**

- (1) Each name specified in the list of names following exclude-word shall be the name of either a RELATIONSHIP-TYPE or RELATIONSHIP-CLASS-TYPE meta-entity in the IRD Schema.
- (2) Duplicates in the list of names shall be ignored.

**Error and Warning Conditions**

- (1) Error E01171: Invalid relationship-type-designator or relationship-class-type-designator. General Rule (1) for this command-clause has been violated.

**6.61 IRD Life-Cycle-Phase Designation Command-Clause**

**Function** To specify an IRD life-cycle-phase.

**Format**

```

IRD-life-cycle-phase-designation-clause ::=
    IRD-word                               4.3
    life-cycle-phase-word                  4.3
    =
    IRD-life-cycle-phase-designator       4.6
    
```

**Syntax Rules**

- (1) IRD-life-cycle-phase-designator shall be defined in the IRD Schema.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01033: Unrecognized IRD life-cycle-phase-designator. Syntax Rule (1) of this command-clause has been violated.

**6.62 Sort Command-Clause**

**Function** To sort the entities selected in an IRD output command into the major sequence for the output display.

**Format**

```

sort-clause ::=
    sort-word                               4.3
    sequence-word                          4.3
    =
    sort-parm-list

sort-parm-list ::=
    sort-parm
    [ , sort-parm-list ]

sort-parm ::=
    sort-parm-form-1
    | sort-parm-form-2

sort-parm-form-1 ::=
    IRD-sort-field

IRD-sort-field ::=
    sorting-IRD-schema-descriptor
    | simple-attribute-type

simple-attribute-type ::=
    attribute-type-designator              4.6

sorting-IRD-schema-descriptor ::=
    life-cycle-phase-word                  4.3
    | entity-type-word                    4.3
    
```

version-word	4.3
variation-word	4.3
revision-word	4.3
entity-access-name-phrase	
entity-descriptive-name-phrase	
entity-access-name-phrase ::=	
[ assigned-word ]	4.3
access-name-word	4.3
entity-descriptive-name-phrase ::=	
[ assigned-word ]	4.3
descriptive-name-word	4.3
sort-param-form-2 ::=	
( IRD-sort-field	
, ascending-or-descending )	
ascending-or-descending ::=	
ascending-word	4.3
descending-word	4.3

**Syntax Rules**     None.

#### **General Rules**

- (1) Each token specified in sort-param-form-1 which is not a sorting-IRD-schema-descriptor shall be the name of an attribute-type.
- (2) Each attribute-type shall be associated with an entity-type.
- (3) For each entity-type associated with a specified attribute-type, the attribute-type shall be nonrepeating.
- (4) If an entity is selected in which the attribute does not exist, (including those cases where the entity-type of the selected entity is not associated with the attribute-type), binary zeroes shall be used for the value of the corresponding sort-field.
- (5) If neither ascending-word nor descending-word is specified, ascending-word shall be assumed.

#### **Error and Warning Conditions**

- (1) Error E01187: Invalid IRD-schema-descriptor used for sort parameter. Either General Rule (1) or (2) has been violated.
- (2) Error E01188: Cannot sort on a repeating attribute. General Rule (3) has been violated.

NOTE: This command-clause corresponds to the sort-meta-entities IRD-schema-clause in Subsection 6.113.

### **6.63 Show Predefined Display Command-Clause**

**Function**     To format a report or query output into a predetermined format.

## Format

```

show-predefined-display-clause ::=
    show-word                               4.3
    display-format

display-format ::=
    standard-display-format-option
    | implementor-defined-format-option

standard-display-format-option ::=
    syntax-format
    | individual-impact-format
    | cumulative-impact-format

syntax-format ::=
    syntax-word                               4.3
    syntax-show-options

individual-impact-format ::=
    [ individual-word ]                       4.3
    impact-word
    impact-show-options

cumulative-impact-format ::=
    cumulative-word                           4.3
    impact-word
    impact-show-options

syntax-show-options ::=
    /* Same as in output-syntax-command */

impact-show-options ::=
    /* Same as in output-impact-of-change-command */

implementor-defined-format-option ::=
    /* implementor-defined-format */

```

## Syntax Rules

- (1) Impact-show-options shall be defined within output-impact-of-change-command.
- (2) Syntax-show-options shall be defined within output-syntax-command.

## General Rules

- (1) The implementor may define additional predefined formats and options for those formats.

**Error and Warning Conditions**    None.

## 6.64 Route Command-Clause

**Function**    To route output to one or more specified destinations.

**Format**

```

route-clause ::=
    route-word           4.3
    to-word              4.3
    destination-list

destination-list ::=
    destination
    [ , destination-list ]

destination ::=
    /* implementor-defined-format */

```

**Syntax Rules**

- (1) The format of any given destination shall be as defined by the implementation.

**General Rules**

- (1) The implementation may limit the number of destinations to a single destination.
- (2) The number of destinations specified shall be less than or equal to the implementation-defined maximum.

**Error and Warning Conditions**

- (1) Error E01189: Invalid destination designation. Syntax Rule (1) has been violated.
- (2) Error E01190: Routing limitations exceeded. General Rule (2) has been violated.

**6.65 Show Title Command-Clause**

**Function** To provide a title which shall appear in the output.

**Format**

```

show-title-clause ::=
    show-word           4.3
    short-string-literal 4.2
    [ first-or-each-page-option ]

first-or-each-page-option
    first-page-option
    | each-page-option

first-page-option ::=
    on-word            4.3
    first-word         4.3
    page-word          4.3

each-page-option ::=
    on-word            4.3

```

each-word 4.3  
page-word 4.3

**Syntax Rules** None.

**General Rules**

- (1) If neither first-page-option nor each-page-option is specified, then each-page-option shall be assumed.
- (2) The title shall be centered at the top of the page.

**Error and Warning Conditions**

- (1) Error E01141: Maximum string length exceeded or string-literal not properly delimited.

**6.66 Entity-Type Show Restriction Command-Clause**

**Function** To restrict the entities displayed on an impact-of-change output to specified entity-types.

**Format**

```

entity-type-show-restriction-clause ::=
    show-word 4.3
    entity-type-option

entity-type-option ::=
    all-option
    | only-entity-types-option

all-option ::=
    all-word 4.3
    entities-word 4.3

only-entity-types-option ::=
    only-word 4.3
    entity-type-list

entity-type-list ::=
    entity-type-designator 4.6
    [ , entity-type-list ]

```

**Syntax Rules** None.

**General Rules**

- (1) Each entity-type-designator shall be the name of an entity-type in the IRD Schema.

**Error and Warning Conditions**

- (1) Error E01002: Invalid or unrecognized entity-type. A name in entity-type-list did not identify a valid entity-type in the IRD Schema.

**6.67 Qualified Show Command-Clause**

**Function** To format output for specific entity-types or relationship-types.

**Format**

```

qualified-show-clause ::=
    entity-type-qualification-clause           6.1
    show-options-within-entity-type

show-options-within-entity-type ::=
    show-clause-options
    | relationship-type-qualified-shows
    | entity-and-relationship-type-qualified-shows

show-clause-options ::=
    show-clause-selection
    [ show-clause-options ]

show-clause-selection ::=
    show-IRD-life-cycle-phase-clause           6.68
    | show-entity-access-name-clause           6.69
    | show-entity-descriptive-name-clause      6.70
    | show-entity-type-clause                  6.71
    | show-relationships-clause                6.72
    | show-attributes-clause                  6.73
    | show-quality-indicator-clause            6.83

relationship-type-qualified-shows ::=
    relationship-type-qualification-clause     6.18
    show-clause-options

entity-and-relationship-type-qualified-shows ::=
    show-clause-options
    relationship-type-qualified-shows
    { relationship-type-qualified-shows }

```

**Syntax Rules**

- (1) There are 3 levels of control within this command-clause. The first level is the entity-type level. The second level shall be used to specify which entity-attributes or relationships are to be shown. The third level shall be reserved for relationship attributes. Any show-clause at the second or third level shall be governed by the qualification-clause at the higher level.
- (2) Any attribute-type or attribute-group-type explicitly identified in any show-clause at level-2 shall be associated with each entity-type identified in the governing entity-type qualification-clause.
- (3) Any relationship-type identified within a relationship-type-qualification-clause or show-clause shall contain at least one entity-type identified in the governing entity-type restriction-clause.

- (4) When relationship-types are identified in a command-clause governed by an entity-type-qualification-clause, only one entity-type shall be specified in that command-clause.
- (5) Any attribute-type or attribute-group-type explicitly identified in any show-clause at level-3 shall be associated with each relationship-type in the governing relationship-type-qualification-clause.

### General Rules

- (1) If a show-clause governed by a relationship-type specifies life-cycle-phase-word, then the IRD life-cycle-phase containing each entity within the relationship shall be displayed.
- (2) If a relationship-type appears as a qualifier, then one of the following conditions shall hold:
  - (a) A show-relationships-clause has specifically identified either the qualifier relationship-type, or the relationship-class-type which is associated with the qualifier relationship-type. Note that if the relationship-type identified in the show-relationships-clause is the inverse of the qualifier relationship-type, this condition shall not be met. The same is true if the show-relationships-clause identifies a class of relationship-types which contain the inverse of the qualifier relationship-type.
  - (b) A show-relationships-clause specified all relationships.
  - (c) A show-relationships-clause specified all forward relationships and the qualifier is a forward relationship-type-designator.
  - (d) A show-relationships-clause specified all inverse relationships and the qualifier is a relationship-type-inverse-name.
- (3) If a show-clause governed by a relationship-type specifies quality-word, then the quality-indicator associated with each entity within the relationship shall be displayed.

### Error and Warning Conditions

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. An entity-type, relationship-type, attribute-type, or attribute-group-type or other IRD-schema-descriptor expected. The designator did not identify a meta-entity.
- (2) Error E01174: Inconsistent IRD-schema-descriptors. General Rule (2), (3), or (5) has been violated.
- (3) Error E01191: Relationship-type must be qualified by a single entity-type. General Rule (4) has been violated.
- (4) Error E01192: Qualifying relationship-type not selected. General Rule (2) has been violated.

### 6.68 Show IRD Life-Cycle-Phase Command-Clause

**Function** To request the displaying of the IRD life-cycle-phase for displayed entities.



**Format**

```

show-IRD-life-cycle-phase-clause ::=
    show-word           4.3
    IRD-word            4.3
    life-cycle-phase-word 4.3

```

**Syntax Rules** None.**General Rules**

- (1) If governed by relationship-type qualification-clause, the IRD life-cycle-phase shall be displayed for each entity.
- (2) If the IRD life-cycle-phase is automatically provided by the requested display, this command-clause shall be ignored.

**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the show-IRD-schema-life-cycle-phase IRD-schema-clause in Subsection 6.130.

**6.69 Show Entity-Access-Name Command-Clause**

**Function** To control the displaying of entity-access-names whenever multiple names shall be displayed.

**Format**

```

show-entity-access-name-clause ::=
    show-word           4.3
    entity-word         4.3
    access-name-display-option

access-name-display-option ::=
    access-name-word    4.3
    | access-name-component-list

access-name-component-list ::=
    access-name-component
    [ , access-name-component-list ]

access-name-component ::=
    assigned-word access-name-word 4.3
    | version-word                 4.3
    | variation-word                4.3
    | revision-word                 4.3

```

**Syntax Rules**

- (1) No access-name-component shall be repeated.
- (2) If version-word is specified, variation-word and revision-word shall not be specified.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01193: Duplicated name component specified. Either Syntax Rule (1) or (2) has been violated.

**6.70 Show Entity-Descriptive-Name Command-Clause**

**Function** To control the displaying of entity-descriptive-names whenever multiple names shall be displayed.

**Format**

```
show-entity-descriptive-name-clause ::=
    show-word           4.3
    entity-word         4.3
    descriptive-name-display-option

descriptive-name-display-option ::=
    descriptive-name-word           4.3
    | descriptive-name-component-list

descriptive-name-component-list ::=
    descriptive-name-component
    [ , descriptive-name-component-list ]

descriptive-name-component ::=
    assigned-word descriptive-name-word  4.3
    | version-word           4.3
    | variation-word         4.3
    | revision-word          4.3
```

**Syntax Rules**

- (1) No descriptive-name-component shall be repeated.
- (2) If version-word is specified, variation-word and revision-word shall not be specified.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01193: Duplicated name component specified. Either Syntax Rule (1) or (2) has been violated.

**6.71 Show Entity-Type Command-Clause**

**Function** To request the displaying of the entity-type for a displayed entity.

**Format**

```

show-entity-type-clause ::=
    show-word                4.3
    entity-type-word         4.3

```

**Syntax Rules** None.**General Rules**

- (1) If this command-clause is governed by relationship-type-qualification-clause, the entity-type of each entity within each displayed relationship shall be displayed.
- (2) This command-clause shall be ignored if the display always provides the information.

**Error and Warning Conditions** None.**6.72 Show Relationships Command-Clause****Function** To control the displaying of relationships within a general-output-command.**Format**

```

show-relationships-clause ::=
    show-word                4.3
    relationship-display-option
    [ attribute-display-option ]

relationship-display-option ::=
    all-relationships-option
    | excluding-relationships-option
    | specified-relationship-option

all-relationships-option ::=
    all-word                4.3
    [ forward-or-inverse-option ]
    relationships-word      4.3

excluding-relationships-option ::=
    all-word                4.3
    relationships-word      4.3
    except-word             4.3
    relationship-type-selection-list

forward-or-inverse-option ::=
    forward-word           4.3
    | inverse-word         4.3

specified-relationships-option ::=
    relationships-word      4.3
    relationship-type-selection-list

```

```

relationship-type-selection-list ::=
    relationship-type-selection
    [ , relationship-type-selection-list ]

relationship-type-selection ::=
    relationship-type-designator           4.6
    | relationship-type-inverse-name       4.6
    | relationship-class-type-designator   4.6
    | relationship-class-type-inverse-name 4.6

attribute-display-option ::=
    and-word                               4.3
    [ all-or-no-option ]
    attributes-word                         4.3

all-or-no-option ::=
    all-word                               4.3
    | no-word                              4.3
    
```

### Syntax Rules

- (1) Each name specified within relationship-type-selection-list shall be a valid name in the IRD Schema for a relationship-type, relationship-class-type, relationship-type-inverse, or relationship-class-type-inverse.
- (2) If a relationship-type, relationship-class-type, or inverse of a relationship-type or relationship-class-type is identified more than once in a command-clause, a warning condition shall be raised. A relationship-class-type-designator or relationship-class-type-inverse-name shall also identify all relationship-types or inverses within the class.

### General Rules

- (1) The specified-attributes-option shall identify whether or not relationship attributes shall be displayed. It can be overridden for a particular relationship-type or inverse by a qualified show-attributes-clause.
- (2) Relationships whose types are ordered shall always be displayed with ORDER or the sequence attribute.
- (3) Forward-or-inverse-option restricts the display to only forward or inverse relationships for any given entity. For any selected entity:
  - (a) The forward option shall indicate that the relationship shall be displayed if it is a forward relationship and the selected entity is entity-1 of the relationship.
  - (b) The inverse option shall indicate that the relationship shall be displayed if it is an inverse relationship and the selected entity is entity-2 of the relationship.
- (4) If all relationships are requested, then for each selected entity:
  - (a) All forward relationships containing the selected entity as entity-1 shall be displayed; and
  - (b) All inverse relationships containing the selected entity as entity-1 shall be displayed.

- (5) If specified-relationships-option is specified, then for each selected entity, each relationship of a type identified within relationship-type-selection-list containing the selected entity as entity-1 shall be shown.
- (6) Exclude-relationships-option identifies that all relationships but those of the specified types shall be displayed.
- (7) For each relationship-type to be shown by this command-clause either all or no attributes shall be displayed according to what is specified in attributes-display-option. The attributes-display-option can be overridden for any given relationship-type by a qualified show-attributes-clause.
- (8) If attribute-display-option is not specified, all attributes for the relationships to be shown by this command-clause shall be displayed unless overridden by a qualified show-attributes-clause.
- (9) A command shall not contain different show-relationships-clauses which identify (either directly or indirectly) the same relationship-type. (Thus it is not possible to specify conflicting attribute-display options for a relationship-type.) In particular, if one show-relationships-clause specifies all-relationship-option, no other show-relationships-clauses shall be specified.

**Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. Syntax Rule (1) of this command-clause has been violated.
- (2) Warning W01030: Duplicate relationship-type. See Syntax Rule (2).
- (3) Error E01194: Relationship-type duplicated in different command-clauses. General Rule (9) has been violated.

NOTE: This command-clause corresponds to the show-meta-relationships IRD-schema-clause in Subsection 6.116.

**6.73 Show Attributes Command-Clause**

**Function** To request the displaying of all or selected attributes and attribute-groups.

**Format**

```

show-attributes-clause ::=
    show-word                4.3
    attribute-selection-option

attribute-selection-option ::=
    all-attributes-phrase
    | no-attributes-phrase
    | only-specified-attributes-phrase
    | specified-attributes-phrase
    | all-attributes-except-phrase
    
```

all-attributes-phrase ::=	
all-word	4.3
attributes-word	4.3
no-attributes-phrase ::=	
no-word	4.3
attributes-word	4.3
only-specified-attributes-phrase ::=	
only-word	4.3
specified-attributes-phrase	
specified-attributes-phrase ::=	
attributes-word	4.3
attribute-selection-list	
all-attributes-except-phrase ::=	
all-word	4.3
attributes-word	4.3
except-word	4.3
attribute-selection-list	
attribute-selection-list ::=	
selected-attribute-or-attribute-group	
[ , attribute-selection-list ]	
selected-attribute-or-attribute-group ::=	
simple-attribute-type	
attribute-group-phrase	
text-attribute-phrase	
simple-attribute-type ::=	
attribute-type-designator	4.6
attribute-group-phrase ::=	
attribute-group-type-designator	4.6
[ ( component-attribute-type-list ) ]	
component-attribute-type-list ::=	
component-attribute-type	
[ , component-attribute-type-list ]	
component-attribute-type ::=	
attribute-type-designator	4.6
text-attribute-phrase ::=	
text-attribute-type	
[ line-range-clause ]	6.21
text-attribute-type ::=	
attribute-type-designator	4.6

## Syntax Rules

- (1) The names used within this command-clause shall identify either an attribute-type or attribute-group-type in the IRD Schema. In particular:
  - (a) Simple-attribute-type here shall be the name of any attribute-type (including audit attributes) which are directly associated with an entity-type or relationship-type and whose PICTURE meta-attribute is not TEXT. In this context, it can be repeating or nonrepeating.
  - (b) Attribute-group-type-designator shall be the name of an attribute-group in the IRD Schema, whether repeating or nonrepeating.
  - (c) Text-attribute-type shall be an attribute-type defined in the IRD Schema with a PICTURE meta-attribute of TEXT.
  - (d) Component-attribute-type shall be the name of an attribute-type which is associated with an attribute-group-type.
- (2) Within attribute-group-phrase, each component-attribute-type shall identify a component-attribute-type of the corresponding attribute-group-type.
- (3) The order of component-attribute-types within attribute-group-phrase need not be the same as the order of the component attribute-types within the identified attribute-group-type.
- (4) Component attribute-types can be specified more than once in an attribute-group-phrase.
- (5) Within any level of qualification (including unqualified show-clauses) no attribute-type or attribute-group-type shall be identified more than once. However, it is possible to identify a given attribute-type or attribute-group-type in different show-clauses for either different qualifiers, or at different levels of qualification.
- (6) For any qualification, only one show-attributes-clause shall be specifiable.

## General Rules

- (1) This command-clause shall cause the displaying of attributes of the types specified. For a repeating attribute, all occurrences shall be displayed.
- (2) If no component-attribute-type-list is specified in parentheses after an attribute-group-type-designator, then all component attributes shall be displayed. If the attribute-group is repeating, then all occurrences shall be displayed in full.
- (3) If component-attribute-type-list is specified after attribute-group-type-designator, then only component attributes of the identified types shall be displayed. Within any one group, the component attributes shall be displayed in the order specified within the component-attribute-type-list.
- (4) If no line-range-clause is specified after text-attribute-type, then all lines of text shall be displayed.
- (5) If a line-range-clause is specified after text-attribute-type, then only lines of text within the specified range shall be displayed.

- (6) A show-attributes-clause at the lowest level of qualification takes precedence except when the command-clause at the lowest level specifies specified-attributes-phrase. In this case the attributes to be displayed shall be the union of all attributes of types specified at both levels. In the event that some attribute-types are identified to be shown at both levels of qualification, the specification at the lowest level takes precedence. Thus:

If the following show-clause is unqualified:

```
SHOW ATTRIBUTES TEXT-1,TEXT-2 (1 THRU 5)
```

and it is followed by the following qualified show-clause:

```
FOR ENTITY-TYPE X SHOW ATTRIBUTES TEXT-1(1),ATTRIB-A
```

then for all entities of type X, the first line of TEXT-1 attributes, the first five lines of TEXT-2 attributes, and ATTRIB-A attributes shall be shown.

Had the unqualified show-clause specified:

```
FOR ENTITY-TYPE X SHOW ONLY TEXT-1(1), ATTRIB-A
```

then only the first line of TEXT-1 and ATTRIB-A would have been shown for all entities of type X.

- (7) All-attributes-except-phrase shall be equivalent to specifying all but the specified attribute-types in attribute-selection-list for each qualifying entity-type or relationship-type.
- (8) If all-attributes-except-phrase is specified, line-range-clauses for associated text-attribute-types shall be ignored. In this case, the entire text-attribute-type shall be ignored, rather than just the specified range of lines. A warning condition shall be raised.

#### **Error and Warning Conditions**

- (1) Error E01126: Unrecognized or invalid IRD-schema-descriptor. See Syntax Rule (1).
- (2) Error E01195: Invalid component attribute-type-designator. See Syntax Rule (2).
- (3) Error E01196: Duplicate or conflicting show of attribute. See Syntax Rules (5) and (6).
- (4) Warning W01031: Entire text-attribute suppressed. See General Rule (8).

NOTE: This command-clause corresponds to the show-meta-attributes IRD-schema-clause in Subsection 6.115.

#### **6.74 Show All Command-Clause**

**Function** To specify that for each selected entity, the following information shall be displayed:

- (1) The entity-type.
- (2) The IRD-partition containing the entity.
- (3) All attributes and attribute-groups.



- (4) All forward relationships where the selected entity is the first entity of the relationship.
- (5) All inverse relationships where the selected entity is the first entity of the relationship.
- (6) All attributes and attribute-groups for the above relationships.

**Format**

```

show-all-clause ::=
    show-word           4.3
    all-word            4.3
    
```

**Syntax Rules**    None.

**General Rules**    None.

**Error and Warning Conditions**    None.

**NOTES:**

- (1) This command-clause corresponds to the IRD-schema-show-all IRD-schema-clause in Subsection 6.114.

**6.75 Show Relationship Syntax Command-Clause**

**Function**    To identify whether relationship syntax should be displayed redundantly for each selected entity, or nonredundantly after all entities.

**Format**

```

show-relationship-syntax-clause ::=
    show-word           4.3
    relationship-word   4.3
    syntax-word         4.3
    relationship-syntax-option
    
```

```

relationship-syntax-option ::=
    for-each-phrase
    | after-last-phrase
    
```

```

for-each-phrase ::=
    for-word           4.3
    each-word         4.3
    [ entity-word ]   4.3
    
```

```

after-last-phrase ::=
    after-word        4.3
    [ last-word       4.3
      [ entity-word ] 4.3
    ]
    
```

**Syntax Rules**    None.

### General Rules

- (1) If for-each-phrase is specified, for each selected entity, all relationships which contain the selected entity and are visible within the IRD-views used by the command, shall be displayed after the entity is displayed. This may result in some relationships being displayed more than once.
- (2) If after-last-phrase is specified, all relationships which contain any selected entity shall be displayed nonredundantly after the last entity is displayed.

**Error and Warning Conditions**     None.

### 6.77 IRDS Function Restriction Command-Clause

**Function**     To restrict the selection of entities or relationships based upon a test involving an IRDS Function. Such tests consist of:

- (1) The length of attributes of a specified type.
- (2) The length of entity assigned-access-names or assigned-descriptive-names.
- (3) The number of lines of text within attributes of a specified type.
- (4) The number of occurrences of a repeating attribute or attribute-group of a specified type.

### Format

```
irds-function-restriction-clause ::=
    irds-function                4.9
    relational-operator          4.3
    unsigned-integer            4.2
```

### Syntax Rules

- (1) For the 'COUNT function, only an attribute-type-designator or attribute-group-type-designator shall be used as a function-qualifier.
- (2) For the 'LENGTH function, if access-name-word or descriptive-name-word is the function-qualifier, then this restriction-clause shall be either unqualified, or governed by an entity-type-qualification-clause.
- (3) If the restriction-clause is governed by an entity-type or relationship-type qualification-clause, the attribute-type-designator or attribute-group-type-designator used as a function-qualifier shall be associated in the IRD Schema with the qualifying entity-type or relationship-type.

**General Rules**     None.

### Error and Warning Conditions

- (1) Error E01174: Inconsistent IRD-schema-descriptors. Either Syntax Rule (2) or Syntax Rule (3) has been violated.

- (2) Error E01167: Invalid use of irds-function. Syntax Rule (1) has been violated.

### 6.78 Entity Assigned-Access-Name Restriction Command-Clause

**Function** To restrict selection of entities based on a test of the assigned-access-name.

**Format**

```
entity-assigned-access-name-restriction-clause ::=
    entity-word           4.3
    assigned-word         4.3
    access-name-word      4.3
    relational-operator    4.3
    irds-name             4.2
```

**Syntax Rules** None.

**General Rules**

- (1) If the assigned-access-name of an entity and the irds-name specified in this command-clause are not of equal length, then the comparison shall treat both names as if they were left-justified character-strings, with the shorter name padded with trailing blanks.

**Error and Warning Conditions** None.

### 6.79 Entity Assigned-Descriptive-Name Restriction Command-Clause

**Function** To restrict selection of entities based on a test of the assigned-descriptive-name.

**Format**

```
assigned-descriptive-name-restriction-clause ::
    entity-word           4.3
    assigned-word         4.3
    descriptive-name-word 4.3
    relational-operator    4.3
    irds-name             4.2
```

**Syntax Rules** None.

**General Rules**

- (1) If the assigned-descriptive-name of an entity and the irds-name specified in this command-clause are not of equal length, then the comparison shall treat both names as if they were left-justified character-strings, with the shorter name padded with trailing blanks.

**Error and Warning Conditions** None.

### 6.80 Variation-Name Restriction Command-Clause

**Function** To restrict selection of entities based on a test of the variation-name.

**Format**

```

variation-name-restriction-clause ::=
    variation-word                4.3
    relational-operator           4.3
    irds-name                    4.2
    
```

**Syntax Rules** None.

**General Rules**

- (1) If the variation-name of an entity and the irds-name specified in this command-clause are not of equal length, then the comparison shall treat both names as if they were left-justified character-strings, with the shorter name padded with trailing blanks.

**Error and Warning Conditions** None.

### 6.81 Revision-Number Restriction Command-Clause

**Function** To restrict selection of entities based on a test of the revision-number.

**Format**

```

revision-number-restriction-clause ::=
    revision-word                4.3
    relational-operator           4.3
    comparison-value

comparison-value ::=
    unsigned-integer            4.2
    | lowest-word                4.3
    | highest-word               4.3
    
```

**Syntax Rules** None.

**General Rules**

- (1) Unless this restriction is connected by an "and" with a variation-name-restriction-clause, lowest-word and highest-word shall mean lowest or highest revision-number associated with the variation-name of the entity being tested.
- (2) The following conditions shall not exist:

```

REVISION < LOWEST
REVISION > HIGHEST
    
```

**Error and Warning Conditions** None.

**6.82 Show Counts Command-Clause**

**Function** To show any or all of the following information:

- (1) The number of entities or relationships selected.
- (2) The number of entities or relationships selected of a particular type.
- (3) The number of occurrences of any repeating attribute.
- (4) The number of lines of text within any text attribute.

**Format**

```

show-counts-clause ::=
    show-word                               4.3
    irds-function-list

irds-function-list ::=
    function-result
    [ , irds-function-list ]

function-result ::=
    count-function-result
    | lines-function-result

count-function-result ::=
    count-function-qualifier'COUNT

count-function-qualifier ::=
    entities-word                           4.3
    | relationships-word                     4.3
    | attributes-word                        4.3
    | entity-type-designator                4.6
    | relationship-type-designator           4.6
    | attribute-type-designator              4.6
    | attribute-group-type-designator        4.6

lines-function-result ::=
    lines-function-qualifier'LINES

lines-function-qualifier ::=
    attributes-word                           4.3
    | text-attribute-type

text-attribute-type ::=
    attribute-type-designator                 4.6

```

**Syntax Rules**

- (1) The irds-function qualifier shall agree with the governing qualification-clause. In particular:
  - (a) In an unqualified show-clause, all function qualifiers shall be allowed.

- (b) If the show-clause is governed by an entity-type-qualification-clause, then entity-type-designator shall not be specified.
  - (c) If the show-clause is governed by a relationship-type-designator, then entities-word, entity-type-designator, and relationship-type-designator shall not be specified.
  - (d) Any particular attribute-type-designator or attribute-group-type-designator shall be associated with the qualifying entity-type or relationship-type.
- (2) Text-attribute-type shall be the name of an attribute-type meta-entity with a PICTURE meta-attribute equal to TEXT.

### General Rules

- (1) If 'COUNT is preceded by entities-word or relationships-word then a count of all entities or relationships selected shall be displayed at end of output. If this command-clause is qualified, then the count shall be displayed for each control break in the qualifying entity-type or relationship-type. If the command-clause is unqualified, then these values shall be shown at the end of output.
- (2) If 'COUNT is preceded by a specific entity-type or relationship-type, then a tally shall be maintained of entities and relationships shown of the specified type. These values shall be displayed at the end of output.
- (3) If attributes-word precedes 'COUNT, then for each attribute or attribute-group, the number of occurrences shall be displayed. If the corresponding attribute-type or attribute-group-type was excluded from display, only these counts shall be shown. Otherwise, these counts shall be shown along with the attribute occurrences.
- (4) If attributes-word precedes 'LINES, then for each text-attribute shown, the total number of lines shall be shown along with the text-attribute. If the text-attribute is excluded from display, or only a subrange of lines are to be shown, the total number of lines within the text-attribute still shall be shown.
- (5) If a specific attribute-type-designator is used as a function-qualifier, then processing shall be as specified in General Rules (3) and (4) above, but only for the specified attribute-type.

### Error and Warning Conditions

- (1) Error E01174: Inconsistent IRD-schema-descriptors. See Syntax Rules (1) and (2).

### 6.83 Show Quality-Indicator Command-Clause

**Function** To show the quality-indicator for an entity.

#### Format

```
show-quality-indicator-clause ::=
    show-word                4.3
    quality-word             4.3
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

### 6.101 Meta-Entity-Type Command-Clause

**Function** To identify the meta-entity-type for an entity in a add-meta-entity-command, and to document the meta-entity-type of an existing meta-entity in a modify-meta-entity-command.

#### Format

```

meta-entity-type-clause ::=
    meta-entity-type-word           4.3
    =
    meta-entity-type                 4.7

```

#### Syntax Rules

- (1) Meta-entity-type shall identify a valid META-ENTITY-TYPE.

**General Rules** None.

#### Error and Warning Conditions

- (1) Error E01197: Invalid meta-entity-type. Syntax Rule (1) for this command-clause has been violated.

NOTE: This command-clause corresponds to the entity-type IRD-clause in Subsection 6.2.

### 6.103 New Meta-Attributes Command-Clause

**Function** To identify those meta-attributes and/or meta-attribute-groups which shall be created within a meta-entity by the add-meta-entity-command or within a meta-relationship by the add-meta-relationship-command.

#### Format

```

new-meta-attributes-clause ::=
    with-word                       4.3
    [ meta-attributes-word ]        4.3
    new-meta-attribute-clause-list

new-meta-attribute-clause-list ::=
    new-meta-attribute-or-meta-attribute-group
    [ new-meta-attribute-clause-list ]

new-meta-attribute-or-meta-attribute-group ::=
    simple-meta-attribute-clause    6.106
    | new-repeating-meta-attribute-clause 6.107
    | new-meta-attribute-group-clause    6.108
    | new-text-meta-attribute-clause     6.109

```

## Syntax Rules

- (1) If this command-clause is associated with a meta-relationship, the sequencing meta-attribute shall not be specified.
- (2) A system-maintained meta-attribute-type (see Section 9) shall not be specified in an add-meta-entity-command or an add-meta-relationship-command. The system shall create the meta-attribute of the corresponding type when required.
- (3) A fixed meta-attribute-type (see Section 9) can be specified within an add-meta-entity-command or an add-meta-relationship-command.
- (4) If a required meta-attribute-type (see Section 9) is not specified within an add-meta-entity-command or an add-meta-relationship-command, a meta-attribute of the corresponding type shall be created with a default value.
- (5) If a meta-attribute-type which is neither system-maintained, fixed, nor required is not specified within an add-meta-entity-command or an add-meta-relationship-command, it shall not be created.

**General Rules** None.

## Error and Warning Conditions

- (1) Error E01044: Invalid, missing, or misplaced command-clause. With-word expected and not recognized or meta-attributes-word misspelled.

NOTE: This command-clause corresponds to the new-entity-attributes IRD-clause in Subsection 6.22.

## 6.104 Modified Meta-Attributes Command-Clause

**Function** To identify those meta-attributes and/or meta-attribute-groups which can be modified, created, or erased within a meta-entity by a modify-meta-entity-command or within a meta-relationship by a modify-meta-relationship-command.

## Format

```

modified-meta-attributes-clause ::=
    with-word                               4.3
    [ meta-attributes-word ]                4.3
    modified-meta-attribute-clause-list

modified-meta-attribute-clause-list ::=
    meta-attribute-or-meta-attribute-group
    [ modified-meta-attribute-clause-list ]

meta-attribute-or-meta-attribute-group ::=
    simple-meta-attribute-clause            6.106
    | modified-repeating-meta-attribute-clause 6.110
    | modified-meta-attribute-group-clause    6.111
    | modified-text-meta-attribute-clause    6.112
    | text-meta-attribute-resequence-clause  6.125
    
```



**Syntax Rules**

- (1) If this command-clause is associated with a meta-relationship, the sequencing meta-attribute shall not be specified.
- (2) A system-maintained meta-attribute-type (see Section 9) shall not be specified in a modify-meta-entity-command or a modify-meta-relationship-command. The system shall maintain meta-attributes of this type as necessary.
- (3) A fixed meta-attribute-type (see Section 9) shall not be changed within a modify-meta-entity-command or a modify-meta-relationship-command.
- (4) A required meta-attribute-type (see Section 9) shall not be deleted within a modify-meta-entity-command or a modify-meta-relationship-command.
- (5) A meta-attribute-type which is neither system-maintained, fixed, nor required can be deleted within a modify-meta-entity-command or a modify-meta-relationship-command.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01044: Invalid, missing, or misplaced command-clause. With-word expected and not recognized or meta-attributes-word misspelled.

NOTE: This command-clause corresponds to the modified-entity-attributes IRD-clause in Subsection 6.23.

**6.106 Simple Meta-Attribute Command-Clause**

**Function** To declare for a meta-entity or a meta-relationship:

- (1) A new nonrepeating meta-attribute, or
- (2) A new value for an existing nonrepeating meta-attribute, or
- (3) The erasing of an existing nonrepeating meta-attribute.

**Format**

```

simple-meta-attribute-clause ::=
    simple-meta-attribute-type           4.7
    =
    meta-attribute                       4.7
    
```

**Syntax Rules**

- (1) The simple-meta-attribute-type shall identify a valid simple META-ATTRIBUTE-TYPE.
- (2) The simple META-ATTRIBUTE-TYPE identified by simple-meta-attribute-type shall be meta-related to the META-ENTITY-TYPE or META-RELATIONSHIP-TYPE which is being created or modified by the command.

- (3) The meta-attribute shall conform to all validation rules which exist for the corresponding META-ATTRIBUTE-TYPE. Consult Section 9 for a description of these validation rules.

### General Rules

- (1) This command-clause shall be specified only once for each META-ATTRIBUTE-TYPE within a command.
- (2) If the meta-attribute is null-mark, then if the meta-attribute exists, it shall be erased. If the meta-attribute does not exist, no action shall be taken.

### Error and Warning Conditions

- (1) Error E01198: Invalid or unrecognized meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01199: Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01200: Invalid meta-attribute or format. Syntax Rule (3) of this command-clause has been violated.
- (4) Error E01201: Meta-attribute or meta-attribute-group specified more than once in command. General Rule (1) of this command-clause has been violated.
- (5) Warning W01032: Meta-attribute does not exist and cannot be erased. A null-mark was specified as a meta-attribute for a meta-attribute which does not exist. See General Rule (2) of this command-clause.

### NOTES:

- (1) If Syntax Rule (3) of this command-clause is violated, the error message(s) shall clearly identify the validation rules which were violated.
- (2) This command-clause corresponds to the simple-attribute IRD-clause in Subsection 6.3.

## 6.107 New Repeating Meta-Attribute Command-Clause

**Function** To declare a repeating meta-attribute for a meta-entity or a meta-relationship when a meta-entity or meta-relationship is created.

### Format

```
new-repeating-meta-attribute-clause ::=
    repeating-meta-attribute-type           4.7
    = repeating-meta-attribute-list         4.7
```

### Syntax Rules

- (1) The repeating-meta-attribute-type shall identify a valid repeating META-ATTRIBUTE-TYPE.

- (2) The repeating META-ATTRIBUTE-TYPE identified by repeating-meta-attribute-type shall be meta-related to the META-ENTITY-TYPE or META-RELATIONSHIP-TYPE which is being created by the command.
- (3) Each meta-attribute in the repeating-meta-attribute-list shall conform to all validation rules which exist for the corresponding META-ATTRIBUTE-TYPE. Consult Section 9 for a description of these validation rules.
- (4) The number of meta-attributes specified within the repeating-meta-attribute-list shall not exceed the number of repetitions allowed for the specified repeating-meta-attribute-type.

#### **General Rules**

- (1) This command-clause shall be specified only once for each META-ATTRIBUTE-TYPE within a command.
- (2) A null-mark shall not be specified as a value in repeating-meta-attribute-list. If a null-mark is the only value specified, a warning shall be issued.
- (3) Each meta-attribute in repeating-meta-attribute-list shall be unique.

#### **Error and Warning Conditions**

- (1) Error E01198: Invalid or unrecognized meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01199: Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01200: Invalid meta-attribute or format. Syntax Rule (3) of this command-clause has been violated.
- (4) Error E01132: Too many values specified. Syntax Rule (4) of this command-clause has been violated.
- (5) Error E01201: Meta-attribute or meta-attribute-group specified more than once in command. General Rule (1) of this command-clause has been violated.
- (6) Warning W01032: Meta-attribute does not exist and cannot be erased. A null-mark was specified as the only meta-attribute for a repeating-meta-attribute which does not exist. See General Rule (2) of this command-clause.
- (7) Error E01149: Duplicated value. General Rule (3) of this command-clause has been violated.

#### **NOTES:**

- (1) If Syntax Rule (3) of this command-clause is violated, the error message(s) shall clearly identify the validation rules which were violated.
- (2) This command-clause corresponds to the new-repeating-attribute IRD-clause in Subsection 6.4.

### 6.108 New Meta-Attribute-Group Command-Clause

**Function** To declare a new meta-attribute-group within an add-meta-entity-command or an add-meta-relationship-command.

**Format**

```

new-meta-attribute-group-clause ::=
    meta-attribute-group-type           4.7
    =
    meta-attribute-group-list           4.7
    
```

**Syntax Rules**

- (1) The meta-attribute-group-type shall identify a valid META-ATTRIBUTE-GROUP-TYPE.
- (2) The META-ATTRIBUTE-GROUP-TYPE identified by meta-attribute-group-type shall be meta-related to the META-ENTITY-TYPE or META-RELATIONSHIP-TYPE which is being created by the command.
- (3) Each meta-attribute-group shall conform to all validation rules in effect for the META-ATTRIBUTE at the corresponding position within the META-ATTRIBUTE-GROUP-TYPE. Consult Section 9 for a description of these validation rules.
- (4) Each meta-attribute-group shall be entered in the order defined for the corresponding META-ATTRIBUTE-GROUP-TYPE.

**General Rules**

- (1) This command-clause shall be specified only once within a command for each META-ATTRIBUTE-GROUP-TYPE.
- (2) Two different meta-attribute-groups shall not have identical sets of required meta-attributes. (The required META-ATTRIBUTE-TYPES of a META-ATTRIBUTE-GROUP-TYPE act like an access key for a record. Thus, this rule states that each meta-attribute-group shall have a unique "key".) Consult Section 9 for a description of each meta-attribute-group-type.

**Error and Warning Conditions**

- (1) Error E01202: Invalid or unrecognized meta-attribute-group-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01199: Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01200: Invalid meta-attribute or format. Syntax Rule (3) of this command-clause has been violated.
- (4) Error E01204: Meta-attributes for meta-attribute-group specified in wrong order. Syntax Rule (4) of this command-clause has been violated.
- (5) Error E01201: Meta-attribute or meta-attribute-group specified more than once in command. General Rule (1) of this command-clause has been violated.

- (6) Error E01205: Duplicate meta-attribute-group key. General Rule (2) of this command-clause has been violated.

## NOTES:

- (1) The error message(s) shall clearly identify:
- (a) Those meta-attribute-group(s) which were erroneous, and
  - (b) Which meta-attributes within each erroneous meta-attribute-group were in error, and which validation rules were violated for each erroneous value.
- (2) This command-clause corresponds to the new-simple-attribute-group IRD-clause in Subsection 6.5.

**6.109 New Text Meta-Attribute Command-Clause**

**Function** To create a meta-attribute which consists of one or more lines of text.

**Format**

```

new-text-meta-attribute-clause ::=
    text-meta-attribute-type           4.7
    [ line-number-defaults ]
    =
    string-literal-list

string-literal-list ::=
    string-literal                     4.2
    [ string-literal-list ]

line-number-defaults ::=
    line-number-defaults-positional-format
    line-number-defaults-non-positional-format

line-number-defaults-positional-format ::=
    ( start [, increment ] )

start ::= unsigned-integer | null      4.2

increment ::= unsigned-integer        4.2

line-number-defaults-non-positional-format ::=
    ( first-default-clause
      [, second-default-clause ] )

first-default-clause ::=
    start-line-number-clause           6.19
    | line-number-increment-clause     6.20

second-default-clause ::=
    start-line-number-clause           6.19
    | line-number-increment-clause     6.20

```

### **Syntax Rules**

- (1) The text-meta-attribute-type shall identify a valid text META-ATTRIBUTE.
- (2) The META-ATTRIBUTE-TYPE identified by text-meta-attribute-type shall be meta-related to the META-ENTITY-TYPE or META-RELATIONSHIP-TYPE which is being created by the command.

### **General Rules**

- (1) This command-clause shall be specified only once for each META-ATTRIBUTE-TYPE within a command.
- (2) If line-number-defaults is not specified, then start and increment shall be assigned implementation-dependent default values.
- (3) Any assigned line number shall not exceed an implementation-dependent maximum value. See subsequent rules for assigning line-numbers.
- (4) The length of a line is implementation-dependent. If a string-literal exceeds line length, system shall then subdivide the string-literal into as many lines-of-text as required, assigning a line number to each.
- (5) Lines of text shall be assigned line numbers in ascending order, incremented by the increment value. Numbering shall begin with the start value. The numbering of lines of text in subsequent string-literals shall begin one increment after the highest line-number assigned in the prior string-literal.
- (6) A null-string ("") shall mean the line does not exist. If specified, the system shall bypass the line-number that would have been assigned to this string-literal, and proceed to the next string-literal.
- (7) If only the null-string is specified, the system shall regard the meta-attribute as not having been declared at all.
- (8) If the assignment of line-numbers according to the start and increment values will result in exceeding the implementation-defined maximum line-number value, then all lines shall be stored, and line-numbers shall be reassigned from the last line backwards. Line numbers reassigned in this fashion shall begin with the maximum line number, and decremented by a value of 1, until such a reassigned line number is greater than the currently assigned line number of the prior line.
- (9) If the nonpositional form of line-number-defaults is used, then start-line-number-clause and line-number-increment-clause each shall be specified only once.

### **Error and Warning Conditions**

- (1) Error E01198: Invalid or unrecognized meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01199: Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship. Syntax Rule (2) of this command-clause has been violated.

- (3) Error E01141: Minimum string length exceeded or string-literal not properly delimited. See General Rule (3) of this command-clause.
- (4) Warning W01021: Only null-string specified. Command-clause ignored. See General Rule (7) of this command-clause.
- (5) Warning W01022: Null-string specified. Line-number skipped. See General Rule (6) of this command-clause.
- (6) Warning W01023: Assigned line numbers will exceed maximum. Reassigning line numbers from back, decrementing by 1. See General Rule (8) of this command-clause.
- (7) Error E01151: Duplicated subclause. Syntax Rule (9) of this command-clause has been violated.

NOTES:

- (1) The maximum number of lines of text shall be sufficiently large to accommodate practical limitations of human input. This value shall be no less than 32767. An implementation may choose a higher value.
- (2) The length of a line of text shall be implementation-dependent. The implementor shall allow for a line length of at least 72 characters.
- (3) This command-clause corresponds to the new-text-attribute IRD-clause in Subsection 6.7.

**6.110 Modified Repeating Meta-Attribute Command-Clause**

**Function** For an existing meta-entity or meta-relationship:

- (1) To create a new repeating meta-attribute.
- (2) To create new values for an existing repeating meta-attribute.
- (3) To replace a value of an existing repeating meta-attribute by another value.
- (4) To erase one or more values of an existing repeating meta-attribute.
- (5) To erase an existing repeating meta-attribute.

**Format**

```

modified-repeating-meta-attribute-clause ::=
    [ all-word-or-new-word ]
    repeating-meta-attribute-type           4.7
    [ meta-attribute-occurrence-qualification ]
    =
    repeating-meta-attribute-list           4.7

all-word-or-new-word ::=
    all-word                               4.3
    | new-word                             4.3
    
```

meta-attribute-occurrence-qualification ::=  
 ( meta-attribute )

4.7

### Syntax Rules

- (1) The repeating-meta-attribute-type shall identify a valid repeating META-ATTRIBUTE-TYPE.
- (2) The META-ATTRIBUTE-TYPE identified by repeating-meta-attribute-type shall be meta-related to the META-ENTITY-TYPE or META-RELATIONSHIP-TYPE which is being modified by the command.
- (3) Each meta-attribute in the repeating-meta-attribute-list shall conform to all validation rules for the corresponding META-ATTRIBUTE-TYPE. Consult Section 9 for a description of these validation rules.
- (4) Null-mark shall not be specified as one of meta-attributes within a repeating-meta-attribute-list.
- (5) If all-word or new-word precedes repeating-meta-attribute-type, then meta-attribute-occurrence-qualification shall not be specified.
- (6) If meta-attribute-occurrence-qualification is specified, then either null-mark or only one meta-attribute shall be specified after the equals-sign.
- (7) If neither all-word nor new-word is specified, and meta-attribute-occurrence-qualification is not specified, all-word shall be assumed.
- (8) Execution of this command-clause shall not result in the maximum number of repetitions being exceeded for meta-attributes of type repeating-meta-attribute-type.

### General Rules

- (1) For each META-ATTRIBUTE-TYPE within a command:
  - (a) If all-word is specified or assumed, no other occurrences of this command-clause shall exist for the specified repeating-meta-attribute-type.
  - (b) If new-word is specified, there can be multiple command-clauses for the specified repeating-meta-attribute-type with meta-attribute-occurrence-qualification also specified.
  - (c) All command-clauses for a given repeating-meta-attribute-type shall be specified consecutively.
- (2) Each meta-attribute specified after the equals-sign in all occurrences of this command-clause within a command shall be unique.
- (3) If repeating-meta-attribute-type does not exist in the meta-entity or meta-relationship being modified, then the command-clause shall be treated as a new-repeating-meta-attribute-clause (Subsection 6.107) from the repeating-meta-attribute-type on. Thus, the command-clause shall be valid even if repeating-meta-attribute-type is preceded by all-word or new-word.



- (4) If repeating-meta-attribute-type identifies a META-ATTRIBUTE-TYPE which exists in the meta-entity or meta-relationship being modified, and meta-attribute-occurrence-qualification is specified, then the meta-attribute specified in meta-attribute-occurrence-qualification shall identify an existing value of the meta-attribute.
- (5) If meta-attribute-type identifies a META-ATTRIBUTE-TYPE which exists in the meta-entity or meta-relationship being modified, and meta-attribute-occurrence-qualification is specified, then:
  - (a) A null-mark shall indicate that the meta-attribute specified in meta-attribute-occurrence-qualification is to be erased.
  - (b) A meta-attribute not equal to null-mark shall indicate that the meta-attribute specified in meta-attribute-occurrence-qualification shall be replaced by the meta-attribute after the equals-sign.
  - (c) The meta-attribute specified after the equals-sign shall not be an existing value of the meta-attribute.
- (6) If all-word is specified or assumed, and repeating-meta-attribute-type exists within the meta-entity or meta-relationship being modified, a null-mark after the equals-sign shall specify that the repeating-meta-attribute-type and all of its values shall be erased.

#### **Error and Warning Conditions**

- (1) Error E01198: Invalid or unrecognized meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01199: Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01200: Invalid meta-attribute or format. Syntax Rule (3) or (4) of this command-clause has been violated.
- (4) Error E01132: Too many values specified. Syntax Rule (8) of this command-clause has been violated.
- (5) Error E01201: Meta-attribute specified more than once in command. General Rule (1) of this command-clause has been violated.
- (6) Error E01149: Duplicated value. General Rule (2) or (5) of this command-clause has been violated.
- (7) Error E01142: Inconsistent command-clause format. Either General Rule (5) or (6) of this command-clause has been violated.
- (8) Error E01143: Current value does not exist. General Rule (4) of this command-clause has been violated.
- (9) Warning W01032: Meta-attribute does not exist and cannot be erased. A null-mark was specified as the only meta-attribute for a repeating meta-attribute which does not exist.
- (10) Warning W01024: All values to be replaced. All-word assumed and repeating-meta-attribute-type exists.

- (11) Warning W01033: Meta-attribute is to be erased. Erasing individual meta-attributes results in the entire meta-attribute being erased.

NOTE: This command-clause corresponds to the modified-repeating-attribute IRD-clause in Subsection 6.8.

### 6.111 Modified Meta-Attribute-Group Command-Clause

**Function** For an existing meta-entity or meta-relationship:

- (1) To create a new meta-attribute-group.
- (2) To create new occurrences of an existing meta-attribute-group.
- (3) To erase all or specific occurrences of a meta-attribute-group.
- (4) To create, modify, or erase specific component meta-attributes within a given occurrence of a meta-attribute-group.

#### Format

```

modified-meta-attribute-group-clause ::=
    [ all-word-or-new-word ]
    meta-attribute-group-type           4.7
    [ existing-group-occurrence ]
    =
    meta-attribute-group-list | null-mark 4.7

all-word-or-new-word ::=
    all-word           4.3
    new-word           4.3

existing-group-occurrence ::=
    meta-attribute-group           4.7
    
```

#### Syntax Rules

- (1) The meta-attribute-group-type shall identify a valid META-ATTRIBUTE-GROUP-TYPE.
- (2) The META-ATTRIBUTE-GROUP-TYPE identified by meta-attribute-group-type shall be meta-related to the META-ENTITY-TYPE or META-RELATIONSHIP-TYPE which is being modified by the command.
- (3) Each meta-attribute specified shall conform to all validation rules for the META-ATTRIBUTE-TYPE at the corresponding position within the META-ATTRIBUTE-GROUP-TYPE. A component meta-attribute can be left undeclared by leaving the corresponding position within the group either blank or null.
  - (a) If the META-ATTRIBUTE-GROUP-TYPE contains  $n$  component META-ATTRIBUTE-TYPES, then there shall be at most  $n-1$  commas within the meta-attribute-group.
  - (b) If there are  $m$  commas, where  $m < n-1$ , the first  $m+1$  component meta-attributes shall be created within the meta-attribute-group in the order defined for the corresponding META-ATTRIBUTE-GROUP-TYPE.

- (4) If existing-group-occurrence is specified, then either null-mark or only one meta-attribute-group shall be specified after the equals-sign.
- (5) Null-mark shall not be specified as one of several meta-attribute-groups.
- (6) If all-word or new-word precedes meta-attribute-group-type, then existing-group-occurrence shall not be specified.
- (7) If neither all-word nor new-word is specified and existing-group-occurrence is not specified, all-word shall be assumed.

### General Rules

- (1) For each META-ATTRIBUTE-GROUP-TYPE within a command:
  - (a) If all-word is specified or assumed, no other occurrences of this command-clause shall exist for the specified meta-attribute-group-type.
  - (b) If new-word is specified, there can be additional command-clauses for the specified meta-attribute-group-type with existing-group-occurrence also specified.
  - (c) All command-clauses for a given meta-attribute-group-type shall be specified consecutively.
- (2) Each meta-attribute-group specified after the equals-sign in all occurrences of this command-clause for a given meta-attribute-group-type shall identify a meta-attribute-group with a unique set of significant component meta-attributes.
- (3) If meta-attribute-group-type does not exist in the meta-entity or meta-relationship being modified, then the command-clause shall be treated as a new-meta-attribute-group-clause (Subsection 6.108) from meta-attribute-group-type on. This implies:
  - (a) The command-clause shall be valid even if meta-attribute-group-type is preceded by all-word or new-word.
  - (b) If all-word is specified it shall be ignored.
  - (c) Each component META-ATTRIBUTE-TYPE of the META-ATTRIBUTE-GROUP-TYPE which is significant shall be specified in each meta-attribute-group with a meta-attribute which is not null-mark or null, and
  - (d) Each nonsignificant component META-ATTRIBUTE-TYPE which has no corresponding value declared shall be regarded as not existing.
- (4) If meta-attribute-group-type identifies a meta-attribute-group which exists in the meta-entity or meta-relationship being modified, and existing-group-occurrence is specified, then the meta-attribute-group specified in existing-group-occurrence shall specify all required meta-attributes within an existing occurrence of the meta-attribute-group. In particular this means:
  - (a) A simple-meta-attribute-clause (Subsection 6.106) shall be specified for each required component META-ATTRIBUTE-TYPE of the META-ATTRIBUTE-GROUP-TYPE. A non-null value shall be given in each position corresponding to a significant META-ATTRIBUTE-TYPE.

- (b) Each value for a required META-ATTRIBUTE-TYPE shall conform to the validation rules for that META-ATTRIBUTE-TYPE.
  - (c) The set of all such meta-attributes in existing-group-occurrence shall match those of an existing occurrence of the meta-attribute-group.
- (5) If meta-attribute-group-type does not exist as a meta-attribute-group in the meta-entity or meta-relationship being modified, then any existing-group-occurrence shall be invalid.
- (6) If meta-attribute-group-type identifies a meta-attribute-group which exists in the meta-entity or meta-relationship being modified, and existing-group-occurrence is specified, then:
- (a) A null-mark indicates that the occurrence specified in existing-group-occurrence shall be erased.
  - (b) A meta-attribute-group not equal to null-mark indicates that the meta-attribute-group specified in existing-group-occurrence shall be replaced by the meta-attribute after the equals-sign.
  - (c) Any and/or all significant meta-attributes of the group can be changed. However, the meta-attribute-group specified after the equals-sign shall not produce a set of significant meta-attributes which duplicate an existing meta-attribute-group, except for the meta-attribute-group occurrence identified by existing-group-occurrence.
- (7) If meta-attribute-group-type does exist in the maintained meta-entity or meta-relationship, and all-word is specified, meta-attribute-group shall be regarded as replacing the existing meta-attribute-group. Meta-attribute-group shall thus be subject to all rules which apply to meta-attribute-group in new-meta-attribute-group-clause. In particular this implies:
- (a) Each component META-ATTRIBUTE-TYPE of the META-ATTRIBUTE-GROUP-TYPE which is significant shall be specified in each attribute-group with an attribute which is not null-mark or null, and
  - (b) Each nonsignificant component META-ATTRIBUTE-TYPE which has no corresponding value declared shall be regarded as not existing.
- (8) If meta-attribute-group-type does exist in the maintained meta-entity or meta-relationship and all-word is not specified, only those component meta-attributes which shall be created, modified, and/or erased need be specified. In particular this means:
- (a) That any component meta-attribute which is not declared shall be left unchanged. If the corresponding component meta-attribute is nonexistent within the group and no value is declared, it remains nonexistent.
  - (b) If null-mark is specified for any nonsignificant component meta-attribute, it shall be declared to be erased if it currently exists, or no change shall be made if it does not exist.
  - (c) Any component meta-attribute with a value declared other than null-mark shall be declared to be created if it does not exist or its value replaced if it does exist within the group.

- (9) If all-word is specified or assumed, and meta-attribute-group-type exists within the meta-entity or meta-relationship being modified, a null-mark after the equals-sign shall identify that all occurrences of meta-attribute-group-type shall be erased.

### Error and Warning Conditions

- (1) Error E01198: Invalid or unrecognized meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01199: Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01133: Missing or unbalanced parentheses. A valid meta-attribute-group format was expected but not recognized.
- (4) Error E01206: Too many repetitions of group. Syntax Rule (4) of this command-clause has been violated.
- (5) Error E01118: Invalid component meta-attribute. Syntax Rule (3) of this command-clause has been violated.
- (6) Error E01152: Too many values within group. Syntax Rule (3) of this command-clause has been violated.
- (7) Error E01205: Duplicate meta-attribute-group key. General Rule (2), (3), or (6) of this command-clause has been violated.
- (8) Error E01142: Inconsistent command-clause format. Syntax Rule (6) of this command-clause has been violated.
- (9) Error E01106: Current meta-attribute-group does not exist. General Rule (4) of this command-clause has been violated.
- (10) Error E01203: Required meta-attribute missing. Syntax Rule (3), (6), or (7) of this command-clause has been violated.
- (11) Error E01145: Incomplete or invalid qualification. General Rule (4) has been violated.
- (12) Error E01207: Meta-attribute-group occurrence does not exist. General Rule (4) has been violated, but there are other occurrences which do exist, and the specified occurrence is not to be erased.
- (13) Warning W01034: Meta-attribute-group cannot be erased. Either:
- (a) Erasure of all occurrences has been requested. No occurrences of meta-attribute-group-type exist within the maintained entity or relationship. Or,
  - (b) General Rule (4) has been violated but the specified occurrence is to be erased.
- (14) Warning W01024: All values to be replaced. All-word assumed and meta-attribute-group-type exists.
- (15) Warning W01035: Meta-attribute-group is to be erased. Erasing individual meta-attribute-group occurrences results in the entire meta-attribute being erased.

NOTES:

- (1) If Syntax Rule (3) of this command-clause is violated, the error message(s) shall clearly identify which meta-attribute(s) or simple-meta-attribute-clause(s) in which group is in error, and which validation rules were violated in each case.
- (2) This command-clause corresponds to the modified-simple-attribute-group IRD-clause in Subsection 6.9.

**6.112 Modified Text Meta-Attribute Command-Clause**

**Function** For an existing meta-entity or meta-relationship:

- (1) To create a text meta-attribute.
- (2) To erase a text meta-attribute.
- (3) To create, modify, or erase any group of one or more lines of text.

**Format**

```

modified-text-meta-attribute-clause ::=
    [ all-word ]                4.3
    text-meta-attribute-type    4.7
    [ line-qualification ]
    =
    line-group | null-mark
    ( subsequent-lines-subclause )

line-qualification ::=
    line-range-clause          6.21
    | line-number-defaults

line-group ::=
    string-literal             4.2
    [ , line-group ]

subsequent-lines-subclause ::=
    lines-word                 4.3
    =
    line-range-clause          6.21
    line-group | null-mark

line-number-defaults ::=
    line-number-defaults-positional-format
    | line-number-defaults-non-positional-format

line-number-defaults-positional-format ::=
    ( start [, increment ] )

start ::= unsigned-integer | null    4.2

increment ::= unsigned-integer      4.2
    
```

```

line-number-defaults-non-positional-format ::=
    ( first-default-clause [,]
      second-default-clause )

first-default-clause ::=
    start-line-number-clause           6.19
    | line-number-increment-clause     6.20

second-default-clause ::=
    start-line-number-clause           6.19
    | line-number-increment-clause     6.20

```

### Syntax Rules

- (1) The text-meta-attribute-type shall identify a valid text META-ATTRIBUTE-TYPE.
- (2) The META-ATTRIBUTE-TYPE identified by text-meta-attribute-type shall be meta-related to the META-ENTITY-TYPE or META-RELATIONSHIP-TYPE which is being created or modified by the command.
- (3) If all-word is specified, line-range-clause shall not be allowed.
- (4) If the nonpositional form of line-number-defaults is used, then start-line-number-clause and line-number-increment-clause each shall be specified only once.

### General Rules

- (1) This command-clause can be specified any number of times in a command for any given text-meta-attribute-type. If specified a multiple number of times within a command, each pair of these command-clauses shall be separated by a text-meta-attribute-resequence-clause (see Subsection 6.125) for the same text-meta-attribute-type.
- (2) If text-meta-attribute-type does not exist within the maintained meta-entity or meta-relationship, all-word shall be assumed if not declared. Line-range-clause shall not be allowed.
- (3) If text-meta-attribute-type exists within the maintained meta-entity or meta-relationship, line-number-defaults shall not be allowed.
- (4) If all-word is specified or assumed and line-number-defaults is not specified, then start and increment shall be assigned implementation-dependent default values.
- (5) The length of a line is implementation-dependent. If a string-literal exceeds line length, system shall then subdivide the string-literal into as many lines-of-text as required, assigning a line number to each.
- (6) Any assigned or specified line number shall not exceed an implementation-dependent maximum value. See subsequent rules for assigning line-numbers.
- (7) If all-word is specified or assumed, then this command-clause shall be identical in format to new-text-meta-attribute-clause from text-meta-attribute-type on, except that a null-mark can be specified after the equals-sign. In particular, this means that no subsequent-lines-subclause shall be specified. The following rules apply in this case:

- (a) If a null-mark is specified after the equals-sign, the meta-attribute shall be designated to be erased. If it does not exist, a warning condition shall exist.
  - (b) If a line-group is specified after the equals-sign, the line-group shall be designated to replace all existing lines, if any. This line group shall be processed exactly as the string-literals of new-text-meta-attribute-clause. See Subsection 6.109 for details.
- (8) If a line-number-range is specified after text-meta-attribute-type, then text-meta-attribute-type shall exist in the maintained meta-entity or meta-relationship. The following rules shall apply in this case and to all subsequent-lines-subclauses:
- (a) If one line number is specified in line-range-clause, then either null-mark or one string-literal no longer than line-length shall be specified to the right of the equals-sign. In this case:
    - (i) A null-mark or null-string shall identify that the line is to be deleted. If it does not exist, a warning condition shall exist.
    - (ii) If the specified line-number does not exist, the string-literal shall be designated to become a new line of text.
    - (iii) If the specified line-number exists, the string-literal shall be designated to replace the existing line in its entirety.
  - (b) If two line-numbers are specified in the line-range-clause, then all existing lines with numbers greater-than or equal-to *line-number-1* and less-than or equal-to *line-number-2* shall be designated to be replaced or erased. In particular, it shall not be necessary that either *line-number-1* or *line-number-2* of the line-range-clause match an existing line-number of text-meta-attribute-type.
    - (i) If null-mark or null-string is specified after the equals-sign, all lines within the range shall be to be deleted. If no lines exist within the specified range, a warning condition shall exist.
    - (ii) If a line-group is specified to the right of the equals-sign, then the line-group shall identify lines which shall replace the existing lines, if any, within the specified range.
    - (iii) The line-group shall not specify more lines than can be placed within the range.
    - (iv) Line-numbers shall be assigned as follows:
      - (A) The total number of lines within the line-group shall be determined. Let  $N$  denote this number.
      - (B) Let  $L$  and  $H$  represent the *line-number-1* and *line-number-2* of the line-range-clause respectively.
      - (C) The increment-value,  $I$ , shall be computed as follows:

$$I = \text{greatest-integer-of } ((H-L)/N).$$



(D) Line numbers shall then be assigned to individual lines, starting with *L* and incremented by *I*.

(9) All line ranges specified in this command-clause shall be disjoint.

**Error and Warning Conditions**

- (1) Error E01198: Invalid or unrecognized meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01199: Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship. Syntax Rule (2) of this command-clause has been violated.
- (3) Error E01141: Maximum string length exceeded or string-literal not properly delimited.
- (4) Error E01153: Inconsistent qualification. Syntax Rule (3) of this command-clause has been violated.
- (5) Error E01108: Duplicated command-clause. General Rule (1) of this command-clause has been violated.
- (6) Error E01151: Duplicated subclause. Syntax Rule (4) of this command-clause has been violated.
- (7) Error E01154: Subsequent lines subclause not allowed. See General Rule (7) of this command-clause.
- (8) Error E01155: String-literal exceeds line length. See General Rule (8) of this command-clause.
- (9) Error E01156: Line-group exceeds specified range. See General Rule (8) of this command-clause.
- (10) Error E01157: Line or range of lines duplicated. General Rule (9) of this command-clause has been violated.
- (11) Warning W01032: Meta-attribute does not exist and cannot be erased.

NOTE: This command-clause corresponds to the modified-text-attribute IRD-clause in Subsection 6.11.

**6.113 Sort Meta-Entities Command-Clause**

**Function** To sort the meta-entities selected in an IRD Schema output command into the major sequence for the output display.

**Format**

```

sort-meta-entities-clause ::=
    sort-word                4.3
    sequence-word            4.3
    =
    IRD-schema-sort-parameter-list
    
```

```

IRD-schema-sort-parameter-list ::=
    IRD-schema-sort-parameter
    [ , IRD-schema-sort-parameter-list ]

IRD-schema-sort-parameter ::=
    IRD-schema-sort-parameter-form-1
    IRD-schema-sort-parameter-form-2

IRD-schema-sort-parameter-form-1 ::=
    meta-entity-type-word           4.3
    | simple-meta-attribute-type    4.7
    | life-cycle-phase-word         4.3

IRD-schema-sort-parameter-form-2 ::=
    ( IRD-schema-sort-parameter-form-1
      , ascending-or-descending-option )

ascending-or-descending-option ::=
    ascending-word                   4.3
    | descending-word                4.3

```

**Syntax Rules**    None.

**General Rules**

- (1) Each token specified in IRD-schema-sort-parameter-form-1 which is not a meta-entity-type-word or life-cycle-phase-word shall be the name of a simple-meta-attribute-type.
- (2) Each meta-attribute-type shall be associated with a meta-entity-type specified for selection.
- (3) If a meta-entity is selected in which the meta-attribute does not exist (including those cases where the meta-entity-type of the selected meta-entity is not associated with the meta-attribute-type), binary zeros shall be used for the value of the corresponding sort-field.
- (4) IRD Schema life-cycle-phases shall be ordered alphabetically.

**Error and Warning Conditions**

- (1) Error E01208: Invalid meta-entity-type used for sort parameter. General Rule (1) of this command-clause has been violated.
- (2) Error E01209: Invalid meta-attribute-type used for sort parameter. General Rule (2) of this command-clause has been violated.

NOTE: This command-clause corresponds to the sort IRD-clause in Subsection 6.62.

**6.114 IRD Schema Show All Command-Clause**

**Function**    To specify that for each selected meta-entity, the following information shall be displayed:

- (1) The meta-entity-type.
- (2) The corresponding IRD Schema life-cycle-phase for each meta-entity.
- (3) All meta-attributes and meta-attribute-groups.
- (4) All meta-relationships in which the selected meta-entity participates.
- (5) All meta-attributes and meta-attribute-groups for the above meta-relationships.

**Format**

```
IRD-schema-show-all-clause ::=
    show-word           4.3
    all-word            4.3
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the show-all IRD-clause in Subsection 6.74.

**6.115 Show Meta-Attributes Command-Clause**

**Function** To request the displaying of all or selected meta-attributes and meta-attribute-groups.

**Format**

```
show-meta-attributes-clause ::=
    show-word           4.3
    all-or-specified-option

all-or-specified-option ::=
    all-meta-attributes-option
    | specified-meta-attributes-option

all-meta-attributes-option ::=
    all-word           4.3
    meta-attributes-word 4.3

specified-meta-attributes-option ::=
    meta-attributes-word 4.3
    meta-attribute-type-list

meta-attribute-type-list ::=
    meta-attribute-or-meta-attribute-group-type
    [ , meta-attribute-type-list ]

meta-attribute-or-meta-attribute-group-type ::=
    non-repeating-meta-attribute-type 4.7
```

repeating-meta-attribute-type	4.7
meta-attribute-group-type	4.7
text-meta-attribute-type-option	
text-meta-attribute-type-option ::=	
text-meta-attribute-type	4.7
[ line-range-clause ]	6.21

**Syntax Rules**

- (1) The names used in the meta-attribute-list within this command-clause shall identify either a valid meta-attribute-type or valid meta-attribute-group-type. In particular:
  - (a) Non-repeating-meta-attribute-type shall identify the name of a non-repeating-meta-attribute-type (including simple-meta-attributes, audit-meta-attributes, static-meta-attributes, and control-meta-attributes) which is associated with a meta-entity-type.
  - (b) Repeating-meta-attribute-type shall identify the name of any valid repeating-meta-attribute.
  - (c) Meta-attribute-group-type shall identify the name of any valid meta-attribute-group.
  - (d) Text-meta-attribute-type shall identify a valid text-meta-attribute-type.

**General Rules**

- (1) This command-clause shall cause the displaying of meta-attributes of the types specified. For a repeating-meta-attribute, all occurrences shall be displayed.
- (2) All component meta-attributes of a specified meta-attribute-group shall be displayed in the same order as the component meta-attributes appear within the corresponding meta-attribute-group-type.
- (3) If no line-range-clause is specified after text-meta-attribute-type, then all lines of text shall be displayed.
- (4) If a line-range-clause is specified after text-meta-attribute-type, then only lines of text within the specified range shall be displayed.
- (5) All-meta-attributes-option shall be equivalent to specifying all meta-attribute-types in specified-meta-attributes-option for each selected meta-entity-type or meta-relationship-type.
- (6) If all-meta-attributes-option is specified, any specified-meta-attributes-option shall be ignored. A warning condition shall be raised.
- (7) If all-meta-attributes-option is specified and a text-meta-attribute-type exists, all lines of text shall be displayed.

**Error and Warning Conditions**

- (1) Error E01211: Unrecognized or invalid meta-attribute-type. Syntax Rule (1) of this command has been violated.

(2) **Warning W01036:** Specified meta-attributes have been ignored. See General Rule (6).

NOTE: This command-clause corresponds to the show-attributes IRD-clause in Subsection 6.73.

### 6.116 Show Meta-Relationships Command-Clause

**Function** To control the displaying of meta-relationships within an Output IRD Schema Command.

**Format**

```

show-meta-relationships-clause ::=
    show-word                               4.3
    meta-relationships-word                 4.3
    [ meta-relationship-meta-attributes-option ]

meta-relationship-meta-attributes-option
    and-word                               4.3
    all-or-no-word
    meta-attributes-word                   4.3

all-or-no-word ::=
    all-word                               4.3
    | no-word                              4.3
    
```

**Syntax Rules** None.

**General Rules**

- (1) For each meta-relationship to be shown, either all or no meta-relationship-attributes shall be displayed according to what is specified in meta-relationship-meta-attributes-option.
- (2) If meta-relationship-meta-attributes-option is not specified, all meta-attributes for the meta-relationships to be shown shall be displayed.

**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the show-relationships IRD-clause in Subsection 6.72.

### 6.117 Show Related Meta-Entities Command-Clause

**Function** To request the display of those meta-entities which are meta-related in some way to selected meta-entities.

**Format**

```

show-related-meta-entities-clause ::=
    show-word                               4.3
    [ directly-or-indirectly-option ]
    related-word
    meta-entities-word                     4.3
    
```

[ where-word 4.3  
 meta-entity-type-restriction-clause ] 6.119

directly-or-indirectly-option ::=  
 directly-word 4.3  
 | indirectly-word 4.3

**Syntax Rules** None.

**General Rules**

- (1) If the directly-or-indirectly-option is not specified, then this command-clause shall display all meta-entities which are either directly or indirectly meta-related to a selected meta-entity.
- (2) If directly-word is specified, only those meta-entities which are directly meta-related to a selected meta-entity shall be displayed.
- (3) If indirectly-word is specified, only those meta-entities which are indirectly meta-related to a selected meta-entity shall be displayed.
- (4) If meta-entity-type-restriction-clause is specified, then only meta-entities of the specified meta-entity-type shall be displayed by this command-clause.

**Error and Warning Conditions** None.

**6.118 Meta-Entity Selection Command-Clause**

**Function** To specify the criteria by which meta-entities are selected.

**Format**

meta-entity-selection-clause ::=  
 select-word 4.3  
 meta-entity-selection-option  
 [ where-word 4.3  
 IRD-schema-restriction-expression ]

meta-entity-selection-option ::=  
 all-word 4.3  
 | meta-entity-name-list

meta-entity-name-list ::=  
 meta-entity-access-name 4.4  
 [ , meta-entity-name-list ]

IRD-schema-restriction-expression ::=  
 IRD-schema-restriction-expression  
 boolean-operator 4.3  
 IRD-schema-restriction-expression  
 | ( IRD-schema-restriction-expression )  
 | [ IRD-schema-descriptor-qualification ]

IRD-schema-restriction-expression	
IRD-schema-restriction-clause	
IRD-schema-descriptor-qualification ::=	
meta-entity-type-qualification-clause	6.120
meta-relationship-type-qualification-clause	6.121
IRD-schema-restriction-clause ::=	
meta-entity-type-restriction-clause	6.119
meta-relationship-existence-	
restriction-clause	6.122
meta-attribute-group-restriction-clause	6.123
meta-attribute-restriction-clause	6.124
IRD-schema-life-cycle-phase-	
restriction-clause	6.129

### Syntax Rules

- (1) Parentheses shall be used to delimit compound expressions. Parentheses define hierarchical levels of control. At any control level, a qualification-clause shall govern that control level and all levels under it. A meta-entity-type-restriction-clause can also govern value and/or substring restriction-clauses which are at lower control levels in the hierarchy.
- (2) A meta-entity-type-qualification-clause shall not be required for an expression restricting values of meta-attributes associated with meta-entities if and only if:
  - (a) Value and/or substring restriction-clauses shall be declared under the control of a meta-entity-type-restriction-clause, and
  - (b) All meta-attribute-types or meta-attribute-group-types specified shall apply to all meta-entity-types within the governing meta-entity-type-restriction-clause.
- (3) Parentheses shall be balanced.
- (4) Any meta-attribute-type or meta-attribute-group-type identified in a restriction-clause shall be associated with (all) the meta-entity-type(s) or meta-relationship-type(s) which govern the command-clause. The meta-entity-type or meta-relationship-type is said to qualify the meta-attribute-type or meta-attribute-group-type.
- (5) If a meta-attribute (as opposed to a scan-pattern) is specified within a restriction-clause, it shall conform to the validation rules of the associated meta-attribute-type.
- (6) A meta-entity-type-qualification-clause shall not be governed by another meta-entity-type-qualification-clause or a meta-relationship-type-qualification-clause or a meta-entity-type-restriction-clause.
- (7) A meta-relationship-existence-restriction-clause or meta-relationship-type-qualification-clause shall be governed by a meta-entity-type-restriction-clause or meta-entity-type-qualification-clause.
- (8) If a meta-relationship-existence-restriction-clause or meta-relationship-type-qualification-clause is specified, meta-entity-type-1 of the meta-relationship-type shall agree with the governing meta-entity-type-restriction-clause or meta-entity-type-qualification-clause.

## General Rules

- (1) If a meta-entity-access-name is specified in meta-entity-selection-clause, and the meta-entity does not exist, an error condition shall occur.
- (2) If additional meta-entity-access-names can specified after a meta-entity-name which is in error by General Rule (1) of this command-clause, processing shall continue on these meta-entities.

## Error and Warning Conditions

- (1) Error E01212: Unrecognized input. A meta-entity-type, meta-relationship-type, meta-attribute-type, or meta-attribute-group-type was expected but not recognized.
- (2) Error E01213: Inconsistent input. A restriction-clause identified a meta-entity-type, meta-relationship-type, meta-attribute-type or meta-attribute-group-type which was not associated with the governing command-clause.
- (3) Error E01153: Inconsistent qualification. One of Syntax Rules (6) through (8) of this command-clause has been violated.
- (4) Error E01166: Nesting levels exceeded in expression. See Note (3).
- (5) Error E01133: Missing or unbalanced parentheses. Syntax Rule (3) of this command-clause has been violated.
- (6) Error E01067: Specified meta-entity does not exist or is not visible. General Rule (1) of this command-clause has been violated.

### NOTES:

- (1) The meta-entity-selection-clause is used to either select specific meta-entities by name, or to select all meta-entities.
- (2) Restriction-clauses then apply tests to the meta-entities obtained via the meta-entity-selection-clause to further narrow retrieval.
- (3) The implementor may restrict the number nesting levels. The minimum shall be 5 levels of nesting.
- (4) This command-clause corresponds to the entity-selection-criteria IRD-clause in Subsection 6.33. Consult the notes for each of the IRD-schema-descriptor-qualification and IRD-schema-restriction-clauses listed above to determine which have corresponding IRD-clauses.

## 6.119 Meta-Entity-Type Restriction Command-Clause

**Function** To restrict a meta-entity-selection-clause to specified meta-entity-types.

### Format

```
meta-entity-type-restriction-clause ::=
    meta-entity-type-word                4.3
    = meta-entity-type-list
```



```

meta-entity-type-list ::=
    meta-entity-type                               4.7
    [ , meta-entity-type-list ]

```

**Syntax Rules**

- (1) Each meta-entity-type shall identify a valid META-ENTITY-TYPE.

**General Rules**

- (1) A meta-entity-type-restriction-clause which specifies  $n$  meta-entity-types shall be equivalent to  $n$  meta-entity-type-restriction-clauses each specifying a single meta-entity-type and connected by OR's.
- (2) Two meta-entity-type-restriction-clauses shall not be connected by an AND.

**Error and Warning Conditions**

- (1) Error E01214: Invalid meta-entity-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01040: Inconsistent command-clauses. General Rule (2) of this command-clause has been violated.

NOTE: This command-clause corresponds to the entity-type-restriction IRD-clause in Subsection 6.34.

**6.120 Meta-Entity-Type Qualification Command-Clause**

**Function** To identify the meta-entity-type(s) which will qualify subsequent restriction-clauses.

**Format**

```

meta-entity-type-qualification-clause ::=
    for-word                                       4.3
    [ meta-entities-word                         4.3
    of-word ]                                     4.3
    meta-entity-type-list

meta-entity-type-list ::=
    meta-entity-type                               4.7
    [ , meta-entity-type-list ]

```

**Syntax Rules**

- (1) Each meta-entity-type shall identify a valid META-ENTITY-TYPE.

**General Rules** None.

**Error and Warning Conditions**

- (1) Error E01214: Invalid meta-entity-type. A meta-entity-type does not identify a valid META-ENTITY-TYPE. Syntax Rule (1) of this command-clause has been violated.

NOTE: This command-clause corresponds to the entity-type-qualification IRD-clause in Subsection 6.1.

### 6.121 Meta-Relationship-Type Qualification Command-Clause

**Function** To identify the meta-relationship-type(s) which will qualify subsequent restriction-clauses.

#### Format

```
meta-relationship-type-qualification-clause ::=
    for-word                               4.3
    meta-relationships-word                4.3
    meta-relationship-type                 4.7
```

#### Syntax Rules

- (1) Each meta-relationship-type shall identify a valid META-RELATIONSHIP-TYPE.

#### General Rules

- (1) For-word shall begin all meta-relationship-type-qualification-clauses.

#### Error and Warning Conditions

- (1) Error E01215: Invalid meta-relationship-type. Syntax Rule (1) of this command-clause has been violated.

NOTE: This command-clause corresponds to the relationship-type-qualification IRD-clause in Subsection 6.18.

### 6.122 Meta-Relationship Existence Restriction Command-Clause

**Function** To restrict selection of meta-entities to those for which:

- (1) Meta-relationships of specified types exist.
- (2) Meta-relationships of specified types do not exist.

#### Format

```
meta-relationships-existence-restriction-clause ::=
    [ no-word ]                             4.3
    meta-relationships-word                 4.3
    meta-relationship-type                 4.7
    exists-word                             4.3
```

#### Syntax Rules

- (1) Each meta-relationship-type shall identify a valid META-RELATIONSHIP-TYPE.

#### General Rules

- (1) This command-clause shall be governed by a meta-entity-type-qualification-clause or a

meta-entity-type-restriction-clause. The governing meta-entity-type shall be the first meta-entity-type of the meta-relationship-type.

- (2) If no-word is specified, then the meta-entity selection shall be restricted to those meta-entities for which meta-relationships of the specified type do not exist.
- (3) If no-word is not specified, then the meta-entity selection shall be restricted to those meta-entities for which meta-relationships of the specified type exist.

**Error and Warning Conditions**

- (1) Error E01127: Improperly qualified command-clause. General Rule (1) of this command-clause has been violated.
- (2) Error E01215: Invalid meta-relationship-type. Syntax Rule (1) of this command-clause has been violated.

NOTE: This command-clause corresponds to the relationship-existence-restriction IRD-clause in Subsection 6.35.

**6.123 Meta-Attribute-Group Restriction Command-Clause**

**Function** To restrict the selection of meta-entities and/or meta-relationships based upon a test of meta-attributes within a meta-attribute-group.

**Format**

```

meta-attribute-group-restriction-clause ::=
    for-word                               4.3
    meta-attribute-group-type              4.7
    component-restriction-expression

component-restriction-expression ::=
    component-restriction-expression
    boolean-operator                       4.3
    component-restriction-expression
    | ( component-restriction-expression )
    | component-meta-attribute-restriction

component-meta-attribute-restriction ::=
    component-meta-attribute-type          4.7
    relational-operator                   4.3
    meta-attribute                         4.7
    
```

**Syntax Rules**

- (1) Meta-attribute-group-type shall be the name of a valid META-ATTRIBUTE-GROUP-TYPE.
- (2) The META-ATTRIBUTE-GROUP-TYPE identified by meta-attribute-group-type shall be meta-related to the qualifying META-ENTITY-TYPE or META-RELATIONSHIP-TYPE.
- (3) The expression shall be terminated by either the command-terminator or a right parentheses which is not balanced by a left parentheses which occurs after meta-attribute-group-type. If any other restriction-clauses are specified after a meta-attribute-group-

restriction-clause, then the entire meta-attribute-group-restriction-clause shall be enclosed in parentheses.

- (4) Each component-meta-attribute-type specified in this command-clause shall identify a META-ATTRIBUTE-TYPE which is meta-related to the META-ATTRIBUTE-GROUP-TYPE identified by meta-attribute-group-type.

**General Rules**

- (1) Component-restriction-expressions connected by and-word shall be assumed to refer to the same occurrence of the meta-attribute-group.

/\* For example, suppose a META-ATTRIBUTE-GROUP-TYPE named MAG has component META-ATTRIBUTE-TYPEs X and Y. Then the restriction-clause:

( FOR MAG X = x1 AND Y = y1 )

is satisfied by the meta-attribute-group:

MAG = (x0,y0)(x1,y1)(x2,y2)

but not by the meta-attribute-group:

MAG = (x0,y1)(x1,y2)(x2,y0).

In order to have both of the above meta-attribute-groups satisfy the restriction, the command-clause would have to be:

(FOR MAG (X = x1 AND Y = y1) OR ((X = x1 AND Y <> y1) AND (X <> x1 AND Y = y1))) \*/

**Error and Warning Conditions**

- (1) Error E01202: Invalid or unrecognized meta-attribute-group-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01088: Meta-relationship does not exist or is not completely visible. Syntax Rule (2) or (4) of this command-clause has been violated.
- (3) Error E01133: Missing or unbalanced parentheses.

NOTE: This command-clause corresponds to the attribute-group-restriction IRD-clause in Subsection 6.37.

**6.124 Meta-Attribute Restriction Command-Clause**

**Function** To restrict the selection of meta-entities and/or meta-relationships based upon a test of meta-attributes.

**Format**

meta-attribute-restriction-clause ::=  
meta-attribute-type

4.7

relational-operator	4.3
meta-attribute	4.7

### Syntax Rules

- (1) Meta-attribute-type shall identify a valid META-ATTRIBUTE-TYPE.
- (2) The META-ATTRIBUTE-TYPE identified by meta-attribute-type shall be associated either with the qualifying META-ENTITY-TYPE or META-RELATIONSHIP-TYPE.

### General Rules

- (1) Null-mark shall be interpreted as a nonexistence indicator. Thus the command-clause "meta-attribute-type = null-mark" shall be interpreted as "meta-attribute-type does not exist". Conversely, the command-clause "meta-attribute-type <> null-mark" means that "meta-attribute-type has been declared."

### Error and Warning Conditions

- (1) Error E01211: Unrecognized or invalid meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01088: Meta-relationship does not exist or is not completely visible. Syntax Rule (2) of this command-clause has been violated.

NOTE: This command-clause corresponds to the attribute-restriction IRD-clause in Subsection 6.36.

## 6.125 Text Meta-Attribute Resequencing Command-Clause

**Function** To specify how to resequence all or a specified range of lines of text of a text meta-attribute.

### Format

```

text-meta-attribute-resequence-clause ::=
    text-meta-attribute-type           4.7
    [ line-range-clause ]             6.21
    =
    resequenced-word                  4.3
    [ start-line-number-clause ]      6.19
    [ line-number-increment-clause ]  6.20

```

### Syntax Rules

- (1) Text-meta-attribute-type shall be the name of a valid text META-ATTRIBUTE-TYPE.

### General Rules

- (1) If no line-range-clause is specified, the range "(FIRST THROUGH LAST)" shall be assumed.
- (2) If start-line-number-clause is not specified, then:

- (a) If all line numbers are to be resequenced, then this command-clause shall have the same default as the starting line number for a new text meta-attribute.
- (b) If a line range is specified, the first line within the range shall retain the same line number.
- (3) If start-line-number-clause is specified, then resequencing of line numbers shall begin with the first line of text within the specified or assumed range. The line number assigned to this first line of text shall be as specified in the start-line-number-clause.
- (4) If the specified or assumed line-range ends with LAST or MAXIMUM, then each line-number after the first line-number within the range shall have a value equal to the prior line number plus the (specified or defaulted) increment value. Line number processing from the initial value of the range shall then be the same as for a new text-meta-attribute.
- (5) If the end of the range is less than the last line number within the text-meta-attribute, then:

Letting  $(L \text{ THROUGH } H)$  represent the specified range,  
let  $N$  represent the number of lines in the range  $(L \text{ THROUGH } H)$ ,  
let  $I$  denote the increment value, and  
let  $S$  be the specified or assumed value for starting line number.

- (a) The range  $(S \text{ THROUGH } S+N)$  shall not include any existing line with a line number outside the range  $(L \text{ THROUGH } H)$ . An error condition shall be raised.
- (b) If line-number-increment-clause is specified, the value of  $I$  shall be as specified if the range  $(S \text{ THROUGH } S+(N-1)*I)$  does not include any line number outside the range  $(L \text{ THROUGH } H)$ .
- (c) If line-number-increment-clause is not specified, the value of  $I$  shall be the same as the default for a new text-meta-attribute if the range  $(S \text{ THROUGH } S+(N-1)*I)$  does not include any line number outside the range  $(L \text{ THROUGH } H)$ .
- (d) If Rule (5)(b) or (5)(c) is violated, then  $I$  shall be computed as follows:  $I = \text{greatest-integer-of } ((H-L)/N)$ . In this case a warning condition shall be raised.

### **Error and Warning Conditions**

- (1) Error E01211: Unrecognized or invalid meta-attribute-type. Syntax Rule (1) of this command-clause has been violated.
- (2) Error E01165: Resequencing line number conflict. General Rule (5)(b) of this command-clause has been violated.
- (3) Warning W01029: Defaulted or specified increment value has been overridden. See General Rule (5)(e) of this command-clause.

NOTE: This command-clause corresponds to the text-attribute-resequence IRD-clause in Subsection 6.29.

**6.126 Meta-Entity Descriptive-Name Command-Clause**

**Function** To identify the assigned-descriptive-name for one or more meta-entities.

**Format**

```

meta-entity-descriptive-name-clause ::=
    meta-entity-word           4.3
    descriptive-name-word      4.3
    =
    assigned-descriptive-name  4.4

```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the entity-descriptive-name-declaration IRD-clause in Subsection 6.27.

**6.127 New Meta-Entity Version Command-Clause**

**Function** To assign a version-identifier to a new entity which has been created by a modify-meta-entity-command or a copy-meta-entity-command.

**Format**

```

new-meta-entity-version-clause ::=
    new-word           4.3
    meta-entity-word   4.3
    version-word       4.3
    [ = version-identifier ] 4.4

```

**Syntax Rules** None.

**General Rules**

- (1) If version-identifier is not specified, the system shall assign a version-identifier to the newly created meta-entity. The variation-name shall be null, and the revision-number shall be one greater than the highest revision-number for any meta-entity with the specified assigned-access-name.
- (2) If version-identifier is null, then:
  - (a) The variation-name shall be null; and
  - (b) The revision-number shall not be equal to the revision-number for any meta-entity with the assigned-access-name specified in the command.

**Error and Warning Conditions**

- (1) E01006: Invalid version-identifier. A version-identifier was expected, but the token did not conform to version-identifier format.

NOTE: This command-clause corresponds to the new-version IRD-clause in Subsection 6.28.

**6.128 With Meta-Relationships Command-Clause**

**Function** In a copy-meta-entity-command, this command-clause specifies that new meta-relationships shall be created. In a delete-meta-entity-command, this command-clause specifies that meta-relationships shall be deleted.

**Format**

with-meta-relationships-clause ::=	
with-word	4.3
meta-relationships-word	4.3

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the with-relationships IRD-clause in Subsection 6.32.

**6.129 IRD Schema Life-Cycle-Phase Restriction Command-Clause**

**Function** To restrict selection of meta-entities based on their corresponding IRD Schema life-cycle-phase.

**Format**

IRD-schema-life-cycle-phase-restriction-clause ::=	
IRD-schema-word	4.3
life-cycle-phase-word	4.3
relational-operator	4.3
IRD-schema-life-cycle-phase	4.7

**Syntax Rules** None.

**General Rules**

(1) For purposes of comparison, IRD Schema life-cycle-phases shall be ordered as follows:

UNCONTROLLED < CONTROLLED < ARCHIVED

**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the IRD-life-cycle-phase-restriction IRD-clause in Subsection 6.46.

**6.130 Show IRD Schema Life-Cycle-Phase Command-Clause**

**Function** To cause the display of the IRD Schema life-cycle-phase corresponding to each meta-entity displayed by an output-IRD-schema-command.



**Format**

```

show-IRD-schema-life-cycle-phase-clause ::=
    show-word                4.3
    IRD-schema-word          4.3
    life-cycle-phase-word    4.3

```

**Syntax Rules** None.**General Rules** None.**Error and Warning Conditions** None.

NOTE: This command-clause corresponds to the show-IRD-life-cycle-phase IRD-clause in Subsection 6.68.

**6.132 Using IRD-Schema-Views Command-Clause****Function** To specify which IRD Schema views are to be used to determine which meta-entities and meta-relationships are visible.**Format**

```

using-IRD-schema-views-clause ::=
    using-word                4.3
    IRD-schema-views-word    4.3
    IRD-schema-view-list

IRD-schema-view-list ::=
    IRD-schema-view-name
    [ , IRD-schema-view-list ]

IRD-schema-view-name ::=
    assigned-access-name

```

**Syntax Rules** None.**General Rules**

- (1) Each IRD-schema-view-name shall be the assigned-access-name of an existing entity of type IRD-SCHEMA-VIEW.
- (2) Each IRD-schema-view specified shall be related to the effective IRDS-user.

**Error and Warning Conditions**

- (1) Error E01116: IRD-schema-view does not exist. IRD-schema-view-name is not the name of an existing entity of type IRD-SCHEMA-VIEW. General Rule (1) of this command-clause has been violated.
- (2) Error E01112: Invalid IRD-schema-view. The specified IRD-schema-view is not related to the effective IRDS-user. General Rule (2) of this command-clause has been violated.

NOTE: This command-clause corresponds to the using-IRD-views IRD-clause in Subsection 6.38.

## 7 Cross Reference Indices

The following subsections present alphabetic indices for the commands and command-clauses found in Section 5 and Section 6 of this standard. A cross-reference index of all error and warning conditions is also provided.

### 7.1 Alphabetic Command Index

<u>Command</u>	<u>Subsection Number</u>
activate-IRD-command	5.1.1.13
add-entity-command	5.2.1.1
add-meta-entity-command	5.1.1.1
add-meta-relationship-command	5.1.1.4
add-relationship-command	5.2.1.4
check-IRD-schema-compatibility-command	5.3.1.3
copy-entity-command	5.2.1.10
copy-meta-entity-command	5.1.1.10
create-IRD-command	5.3.1.1
deactivate-IRD-command	5.1.1.11
delete-entity-command	5.2.1.3
delete-meta-entity-command	5.1.1.3
delete-meta-relationship-command	5.1.1.6
delete-relationship-command	5.2.1.6
enter-panel-dialogue-command	5.3.2.5
exit-IRDS-system-command	5.3.2.4
export-IRD-command	5.3.1.2
general-output-command	5.2.2.1
help-command	5.3.2.3
import-IRD-command	5.3.1.4
IRD-schema-output-command	5.1.2
modify-entity-access-name-command	5.2.1.7
modify-entity-descriptive-name-command	5.2.1.8
modify-entity-command	5.2.1.2
modify-entity-life-cycle-phase-command	5.2.1.9
modify-meta-entity-access-name-command	5.1.1.7
modify-meta-entity-command	5.1.1.2
modify-meta-entity-descriptive-name-command	5.1.1.8
modify-meta-entity-life-cycle-phase-command	5.1.1.9
modify-meta-relationship-command	5.1.1.5
modify-relationship-command	5.2.1.5
output-impact-of-change-command	5.2.2.2
output-syntax-command	5.2.2.3

restore-IRD-schema-command	5.1.1.12
session-status-command	5.3.2.2
set-session-defaults-command	5.3.2.1

## 7.2 Alphabetic Command-Clause Index

<u>Command-Clause</u>	<u>Subsection Number</u>
all-entities-command-clause	6.42
alternate-name-restriction-command-clause	6.49
attribute-group-restriction-command-clause	6.37
attribute-restriction-command-clause	6.36
audit-attribute-restriction-command-clause	6.47
entity-access-name-selection-command-clause	6.12
entity-assigned-access-name-restriction-command-clause	6.78
entity-assigned-descriptive-name-restriction-command-clause	6.79
entity-descriptive-name-declaration-command-clause	6.27
entity-descriptive-name-selection-command-clause	6.41
entity-selection-criteria-command-clause	6.33
entity-type-command-clause	6.2
entity-type-qualification-command-claus	6.1
entity-type-restriction-command-clause	6.34
entity-type-show-restriction-command-clause	6.66
exclude-relationships-of-type-command-clause	6.60
file-title-suffix-command-clause	6.58
in-file-command-clause	6.52
in-IRD-command-clause	6.51
IRD-export-file-command-clause	6.56
IRD-life-cycle-phase-designation-command-clause	6.61
IRD-life-cycle-phase-restriction-command-clause	6.46
IRD-schema-export-file-command-clause	6.55
IRD-schema-life-cycle-phase-restriction-command-clause	6.129
IRD-schema-show-all-command-clause	6.114
IRD-schema-source-command-clause	6.53
irds-function-restriction-command-clause	6.77
line-number-increment-command-clause	6.20
line-range-command-clause	6.21
load-IRD-command-clause	6.54
location-command-clause	6.57
meta-attribute-group-restriction-command-clause	6.123
meta-attribute-restriction-command-clause	6.124
meta-entity-descriptive-name-command-clause	6.126
meta-entity-selection-command-clause	6.118
meta-entity-type-command-clause	6.101
meta-entity-type-qualification-command-clause	6.120
meta-entity-type-restriction-command-clause	6.119
meta-relationship-existence-restriction-command-clause	6.122

meta-relationship-type-qualification-command-clause	6.121
modified-entity-attributes-command-clause	6.23
modified-meta-attributes-command-clause	6.104
modified-meta-attribute-group-command-clause	6.111
modified-relationship-attributes-command-clause	6.44
modified-repeating-attribute-command-clause	6.8
modified-repeating-attribute-group-command-clause	6.10
modified-repeating-meta-attribute-command-clause	6.110
modified-simple-attribute-group-command-clause	6.9
modified-text-attribute-command-clause	6.11
modified-text-meta-attribute-command-clause	6.112
new-entity-attributes-command-clause	6.22
new-meta-attributes-command-clause	6.103
new-meta-attribute-group-command-clause	6.108
new-meta-entity-version-command-clause	6.127
new-relationship-attributes-command-clause	6.43
new-relationship-identification-command-clause	6.25
new-repeating-attribute-command-clause	6.4
new-repeating-attribute-group-command-clause	6.6
new-repeating-meta-attribute-command-clause	6.107
new-simple-attribute-group-command-clause	6.5
new-text-attribute-command-clause	6.7
new-text-meta-attribute-command-clause	6.109
new-version-command-clause	6.28
order-command-clause	6.17
order-restriction-command-clause	6.26
other-IRD-schema-command-clause	6.59
qualified-show-command-clause	6.67
quality-indicator-designation-command-clause	6.30
quality-indicator-restriction-command-clause	6.31
related-entities-command-clause	6.48
relationship-existence-restriction-command-clause	6.35
relationship-identification-command-clause	6.24
relationship-selection-command-clause	6.45
relationship-type-qualification-command-clause	6.18
relationship-type-restriction-command-clause	6.50
revision-number-restriction-command-clause	6.81
route-command-clause	6.64
show-entity-access-name-command-clause	6.69
show-all-command-clause	6.74
show-attributes-command-clause	6.73
show-counts-command-clause	6.82
show-IRD-life-cycle-phase-command-clause	6.68
show-entity-descriptive-name-command-clause	6.70
show-entity-type-command-clause	6.71
show-meta-attributes-command-clause	6.115
show-meta-relationships-command-clause	6.116
show-predefined-display-command-clause	6.63
show-quality-indicator-command-clause	6.83

show-related-meta-entities-command-clause	6.117
show-relationships-command-clause	6.72
show-relationship-syntax-command-clause	6.75
show-IRD-schema-life-cycle-phase-command-clause	6.130
show-title-command-clause	6.65
simple-attribute-command-clause	6.3
simple-meta-attribute-command-clause	6.106
sort-command-clause	6.62
sort-meta-entities-command-clause	6.113
start-line-number-command-clause	6.19
text-attribute-resequence-command-clause	6.29
text-attribute-substring-restriction-command-clause	6.40
text-meta-attribute-resequence-command-clause	6.125
using-IRD-views-command-clause	6.38
using-IRD-schema-views-command-clause	6.132
variation-name-restriction-command-clause	6.80
with-meta-relationships-command-clause	6.128
with-relationships-command-clause	6.32

### 7.3 Error and Warning Condition Index

<u>Error/Warning Condition</u>	<u>Message/Subsection(s)</u>
E01001	Unrecognized or invalid command format. 5
E01002	Invalid or unrecognized entity-type. 5.2.1.1 6.1 6.2 6.25 6.34 6.66
E01004	User designated entity-access-name required. 5.2.1.1 5.2.1.10 6.25
E01005	Invalid name format. 5.2.1.1 5.2.1.7 6.12 6.41
E01006	Invalid version-identifier. 5.2.1.1 6.28 6.127
E01007	Entity-access-name conflict. 4.4 5.2.1.7
E01009	Entity does not exist or is not visible. 4.4 5.2.1.2 5.2.1.3 5.2.1.7 5.2.1.8 5.2.1.9 6.12 6.25 6.41 6.45 6.48
E01010	Entity is not of specified entity-type. 5.2.1.2

- E01013 Entity is related to other entities.  
5.2.1.3
  
- E01017 Relationship already exists.  
5.2.1.4 6.25
  
- E01018 Relationship does not exist.  
5.2.1.5 5.2.1.6 6.24
  
- E01022 Multiple versions not allowed for entities of entity-type.  
5.2.1.1 5.2.1.2 5.2.1.10
  
- E01031 Entity-descriptive-name conflict.  
5.2.1.8 6.27
  
- E01032 Invalid entity-descriptive-name.  
5.2.1.8 6.27
  
- E01033 Unrecognized IRD life-cycle-phase-designator.  
5.2.1.9 6.46 6.61
  
- E01034 Invalid phase transition.  
5.2.1.9
  
- E01035 Entity not in specified phase.  
5.2.1.9
  
- E01036 New-entity-access-name cannot be the same as existing-entity-access-name.  
5.2.1.10
  
- E01037 Descriptive-name cannot be assigned.  
5.2.1.10
  
- E01039 Invalid IRD name.  
5.3.1.1 6.51
  
- E01040 Inconsistent command-clauses.  
5.2.2 5.2.2.1 5.2.2.2 5.3.1.1 6.34 6.35  
6.119
  
- E01041 New-IRD-name already exists.  
5.3.1.1
  
- E01042 Access to IRD Schema denied.  
5.3.1.1 5.3.1.4
  
- E01043 IRD-schema-export-file is not synchronized with IRD-export-file.  
5.3.1.1 5.3.1.4
  
- E01044 Invalid, missing, or misplaced command-clause.  
5.2.2 5.3.1.2 5.3.2.1 6.22 6.23 6.24  
6.25 6.43 6.44 6.51 6.52 6.54  
6.55 6.56 6.103 6.104

- E01045 The IRD-schema-export-file-name specified in the IRD-schema-export-file-command-clause already exists.  
5.3.1.2
- E01046 The IRD-export-file-name specified in the IRD-export-file-command-clause already exists.  
5.3.1.2
- E01048 Access to other IRD Schema denied.  
5.3.1.3
- E01049 Meta-entity exists in source-IRD Schema but not in target-IRD Schema.  
5.3.1.3
- E01050 Meta-relationship exists in source-IRD Schema but not in target-IRD Schema.  
5.3.1.3
- E01051 An attribute-type or attribute-group-type is NO in the source-IRD Schema but YES in target-IRD Schema.  
5.3.1.3
- E01052 Source-IRD Schema DATA-RANGE is not a subrange of corresponding target-IRD Schema DATA-RANGE.  
5.3.1.3
- E01053 Source-IRD Schema meta-attribute does not exist in target-IRD Schema.  
5.3.1.3
- E01054 Source-IRD Schema meta-attribute must be greater than or equal to corresponding target-IRD Schema meta-attribute.  
5.3.1.3
- E01055 Source-IRD Schema meta-attribute must be less than or equal to corresponding target-IRD Schema meta-attribute.  
5.3.1.3
- E01056 Incompatible IRD Schemas.  
5.3.1.4
- E01057 The IRD-schema-export-file-name specified in the IRD-schema-export-file-command-clause does not exist.  
5.3.1.4
- E01058 The IRD-export-file specified in the IRD-export-file-command-clause does not exist.  
5.3.1.4
- E01059 Descriptive-name may not be changed by a Modify Entity Command.  
5.2.1.2
- E01060 Cannot import into specified IRD life-cycle-phase.  
5.3.1.4
- E01061 Source-IRD Schema and target-IRD Schema meta-attributes not identical.  
5.3.1.3

- E01062    User has no IRD-schema-view.  
5.1
- E01063    Effective IRD-schema-view not associated with correct IRD Schema life-cycle-phase.  
5.1.1    5.1.1.1    5.1.1.2    5.1.1.4    5.1.1.5    5.1.1.6  
5.1.1.9    5.1.1.10
- E01064    Meta-entity cannot be modified.  
5.1.1.2
- E01065    Meta-entity already exists.  
4.4
- E01066    Meta-entity assigned-access-name conflicts with meta-entity-substitute-name.  
4.4
- E01067    Meta-entity does not exist or is not visible.  
4.4    5.1.1    5.1.1.2    5.1.1.4    5.1.1.9    5.1.1.10  
6.118
- E01068    Meta-entity assigned-access-name conflicts with inverse-name.  
4.4
- E01069    Invalid meta-entity name format.  
4.4
- E01070    Meta-entity assigned-access-name conflicts with assigned-descriptive-name of an  
existing meta-entity.  
4.4
- E01071    Meta-entities of specified type are system created.  
5.1.1.1
- E01072    Meta-entities of specified type must be created in CONTROLLED IRD Schema life-  
cycle-phase.  
5.1.1.1
- E01073    Meta-entity is locked and cannot be deleted.  
5.1.1.3
- E01074    Meta-entity is not of specified type.  
5.1.1.2
- E01075    Meta-entity already has a descriptive-name.  
5.1.1.2    5.1.1.10
- E01076    Meta-entity cannot be modified. New version is required.  
5.1.1.2
- E01077    New version is not allowed for specified meta-entity.  
5.1.1.2



- E01078 Meta-entity cannot be deleted. Specified meta-entity is related to another meta-entity.  
5.1.1.3
- E01079 Cannot delete specified meta-entity. Other meta-entities depend on specified meta-entity.  
5.1.1.3
- E01080 Invalid meta-relationship-type.  
5.1.1.4
- E01081 Meta-relationship already exists.  
5.1.1.4
- E01082 Conflicting sequencing meta-attributes.  
5.1.1.4
- E01083 Meta-relationship violates ratio constraint.  
5.1.1.4
- E01084 MAXIMUM-NUMBER-OF-OCCURRENCES and SINGULAR meta-attributes conflict.  
5.1.1.4 5.1.1.5
- E01085 Cannot establish meta-relationship. First meta-entity not associated with an ATTRIBUTE-TYPE-VALIDATION-PROCEDURE.  
5.1.1.4
- E01086 Cannot establish meta-relationship. Specified ATTRIBUTE-TYPE-VALIDATION-DATA not consistent with ATTRIBUTE-TYPE-VALIDATION-PROCEDURE associated with first meta-entity.  
5.1.1.4
- E01087 Creation of meta-relationship would violate IRD Schema life-cycle-phase-integrity.  
5.1.1.4
- E01088 Meta-relationship does not exist or is not completely visible.  
5.1.1.5 5.1.1.6 6.123 6.124
- E01089 Meta-relationship maintenance would impact a nonmodifiable meta-entity.  
5.1.1.5 5.1.1.6
- E01090 Maintenance would result in an inconsistency between the IRD and the IRD Schema.  
5.1.1.6
- E01091 Meta-relationship is locked and cannot be deleted.  
5.1.1.6
- E01092 No meta-entity exists with specified name.  
5.1.1.7 5.1.1.8
- E01093 New name conflicts with an assigned-access-name of an existing meta-entity.  
5.1.1.7 5.1.1.8 5.1.1.10

- E01094 New name conflicts with an assigned-descriptive name of existing meta-entity.  
5.1.1.7 5.1.1.8 5.1.1.10
- E01095 The file-format of the IRD-schema-export-file and/or IRD-export-file is not recognized.  
5.3.1.4
- E01096 New meta-entity-version-clause not allowed for specified meta-entity.  
5.1.1.10
- E01097 Assigned-access-name is reserved and cannot be changed.  
5.1.1.7
- E01098 IRD must be deactivated.  
5.1.1.7 5.1.1.8 5.1.1.9 5.1.1.12 5.1.1.13
- E01099 Another user deactivated the IRD.  
5.1.1.7 5.1.1.8 5.1.1.9 5.1.1.12 5.1.1.13
- E01100 Invalid phase transition.  
5.1.1.9
- E01101 Meta-entity not in current IRD Schema life-cycle-phase.  
5.1.1.9
- E01102 Meta-entity is of type which is not valid for command.  
5.1.1.9 5.1.1.10
- E01103 Meta-entity cannot be moved. It would result in meta-relationships which violate IRD Schema life-cycle-phase integrity.  
5.1.1.9
- E01104 IRD is already deactivated.  
5.1.1.10
- E01105 Invalid relationship-type-designator or inverse.  
6.18 6.35
- E01106 Current meta-attribute-group does not exist.  
6.111
- E01107 Duplicate qualification.  
5.2.2.2
- E01108 Duplicated command-clause.  
5.2.2.1 5.2.2.2 5.2.2.3 5.3.2.1 6.11 6.112
- E01109 Entities of specified entity-type will not be shown.  
5.2.2.2
- E01110 Command-clause option invalid for command.  
5.2.2.3

- E01111 Invalid IRD-view.  
5.3.2.1
- E01112 Invalid IRD-schema-view.  
5.3.2.1
- E01113 Cannot establish meta-relationship.  
5.1.1.4
- E01114 Assigned-access-name cannot be changed.  
5.1.1.7
- E01115 Effective IRD-view not associated with correct IRD-partition.  
5.2.1
- E01116 IRD-schema-view does not exist.  
6.132
- E01118 Invalid component meta-attribute.  
6.111
- E01120 Entity not connectable via specified path(s).  
6.48
- E01121 Invalid or unrecognized option.  
5.3.2.2
- E01122 Duplicate option.  
5.3.2.2
- E01123 Incorrect command-imperative.  
5.3.2.5
- E01124 Command is not supported by Panel Interface.  
5.3.2.5
- E01125 IRD-View does not identify a valid IRD-Partition.  
5.2          6.38
- E01126 Unrecognized or invalid IRD-schema-descriptor.  
6.1          6.18          6.29          6.33          6.36          6.37  
6.40          6.45          6.47          6.50          6.67          6.72  
6.73
- E01127 Improperly qualified command-clause.  
6.1          6.18          6.35          6.50          6.122
- E01128 Invalid attribute or format.  
6.3          6.4          6.8          6.49
- E01129 Attribute-type specified more than once in command.  
6.3          6.4          6.7          6.8

E01130	Declaration of value for audit attribute not allowed.					
	6.3					
E01131	Repeating attribute specified.					
	6.3	6.5	6.9			
E01132	Too many values specified.					
	6.4	6.5	6.6	6.8	6.9	6.10
	6.36	6.47	6.107	6.110		
E01133	Missing or unbalanced parentheses.					
	6.5	6.6	6.9	6.10	6.33	6.37
	6.45	6.111	6.118	6.123		
E01134	Invalid component attribute.					
	6.5	6.6	6.9	6.10		
E01135	Unrecognized component attribute-type.					
	6.5	6.6	6.9	6.10		
E01136	Inconsistent component attribute-type.					
	6.5	6.6	6.9	6.10		
E01137	Required attribute missing.					
	6.5	6.9	6.10			
E01138	Required component attribute not specified.					
	6.6					
E01139	Attribute-group duplicated.					
	6.6	6.10				
E01140	Nonrepeating attribute specified.					
	6.6	6.10				
E01141	Maximum string length exceeded or string-literal not properly delimited.					
	6.7	6.11	6.65	6.109	6.112	
E01142	Inconsistent command-clause format.					
	6.8	6.10	6.45	6.110	6.111	
E01143	Current value does not exist.					
	6.8	6.110				
E01144	Current attribute-group does not exist.					
	6.10					
E01145	Incomplete or invalid qualification.					
	6.10	6.111				
E01146	Attribute-group occurrence does not exist.					
	6.10					

E01147	Invalid or unrecognized attribute-type-designator.	6.3	6.4	6.5	6.6	6.7	6.8
		6.9	6.10	6.11			
E01148	Attribute not consistent with entity or relationship.	6.3	6.4	6.7	6.8	6.10	6.11
E01149	Duplicated value.	6.4	6.8	6.107	6.110		
E01150	Attribute-group not consistent with entity or relationship.	6.5	6.6	6.9			
E01151	Duplicated subclause	6.7	6.11	6.109	6.112		
E01152	Too many values within group.	6.5	6.6	6.9	6.10	6.111	
E01153	Inconsistent qualification.	6.11	6.33	6.112	6.118		
E01154	Subsequent lines subclause not allowed.	6.11	6.112				
E01155	String-literal exceeds line length.	6.11	6.112				
E01156	Line-group exceeds specified range.	6.11	6.112				
E01157	Line or range of lines duplicated.	6.11	6.112				
E01158	Variation-name does not exist.	6.12	6.41				
E01159	Invalid line number.	6.19	6.21				
E01160	Invalid increment value.	6.20					
E01161	Relationship and entity-1-name do not exist.	6.24					
E01162	Relationship from entity-1-name to entity-2-name does not exist.	6.24					
E01163	Incorrect ORDER or sequence-attribute.	6.24					
E01164	Duplicated relationship.	6.25					

- E01165    Resequencing line number conflict.  
6.29        6.125
- E01166    Nesting levels exceeded in expression.  
6.33        6.118
- E01167    Invalid use of irds function.  
6.33        6.45        6.77
- E01168    Inconsistent test-value and relational operator.  
6.36        6.47
- E01169    Invalid test-value.  
6.36        6.47
- E01170    IRD-View does not exist.  
6.38
- E01171    Invalid relationship-type-designator or relationship-class-type-designator.  
6.24        6.25        6.60
- E01172    Missing order command-clause or sequence attribute or incorrect sequence attribute.  
6.24        6.25
- E01173    Specified entity-type is inconsistent.  
6.25
- E01174    Inconsistent IRD-schema-descriptors.  
6.26        6.33        6.36        6.37        6.45        6.47  
6.67        6.77        6.82
- E01175    Invalid quality-indicator-designator.  
6.30        6.31
- E01176    Inconsistent subclause.  
6.40
- E01177    Invalid order value.  
6.44
- E01178    Invalid path specified.  
6.48
- E01180    Command-clause(s) not consistent for all relationship-types in relationship-class-type.  
6.50
- E01181    Identified IRD does not exist at specified or assumed location.  
6.51
- E01182    Invalid file-name.  
6.52        6.54

- E01183 Specified file does not exist at specified or assumed location.  
6.52          6.54
- E01184 Invalid file format.  
6.52          6.54
- E01185 Invalid IRD-schema-export-file-name.  
6.55
- E01186 Invalid IRD-export-file-name.  
6.56
- E01187 Invalid IRD-schema-descriptor used for sort parameter.  
6.62
- E01188 Cannot sort on a repeating attribute.  
6.62
- E01189 Invalid destination designation.  
6.64
- E01190 Routing limitations exceeded.  
6.64
- E01191 Relationship-type must be qualified by a single entity-type.  
6.67
- E01192 Qualifying relationship-type not selected.  
6.67
- E01193 Duplicated name component specified.  
6.69          6.70
- E01194 Relationship-type duplicated in different command-clauses.  
6.72
- E01195 Invalid component attribute-type-designator.  
6.73
- E01196 Duplicate or conflicting show of attribute.  
6.73
- E01197 Invalid meta-entity-type.  
6.101
- E01198 Invalid or unrecognized meta-attribute-type.  
6.106      6.107      6.109      6.110      6.111      6.112
- E01199 Meta-attribute or meta-attribute-group not consistent with meta-entity or meta-relationship.  
6.106      6.107      6.108      6.109      6.110      6.111      6.112
- E01200 Invalid meta-attribute or format.  
6.106      6.107      6.108      6.110

- E01201    Meta-attribute or meta-attribute-group specified more than once in command.  
6.106      6.107      6.108      6.110
- E01202    Invalid or unrecognized meta-attribute-group-type.  
6.108      6.123
- E01203    Required meta-attribute missing.  
6.111
- E01204    Meta-attributes for meta-attribute-group specified in wrong order.  
6.108
- E01205    Duplicate meta-attribute-group key.  
6.108      6.111
- E01206    Too many repetitions of group.  
6.111
- E01207    Meta-attribute-group occurrence does not exist.  
6.111
- E01208    Invalid meta-entity-type used for sort parameter.  
6.113
- E01209    Invalid meta-attribute-type used for sort parameter.  
6.113
- E01211    Unrecognized or invalid meta-attribute-type.  
6.115      6.124      6.125
- E01212    Unrecognized input.  
6.118
- E01213    Inconsistent input.  
6.118
- E01214    Invalid meta-entity-type.  
6.119      6.120
- E01215    Invalid meta-relationship-type.  
6.121      6.122
- E01216    Declaration of attribute for audit attribute-group-type not allowed.  
6.5        6.9
- E01217    Attribute-group-type specified more than once in command.  
6.5        6.9
- E01900    Two meta-entities in CONTROLLED phase have the same assigned-access-name.  
5.1.1.13
- E01901    Incomplete relationship-type.  
5.1.1.13



- E01902 Unused meta-entity in CONTROLLED phase.  
5.1.1.13
- E01903 Attribute-group-type has an incomplete key.  
5.1.1.13
- E01904 Two CONTROLLED meta-entities have the same substitute-name.  
5.1.1.13
- E01905 Two CONTROLLED meta-entities have the same inverse-name.  
5.1.1.13
- E01906 Inverse-name of CONTROLLED meta-entity is same as the substitute-name of a CONTROLLED meta-entity.  
5.1.1.13
- E01907 Inverse-name of CONTROLLED meta-entity is the same as the assigned-access-name of a CONTROLLED meta-entity.  
5.1.1.13
- E01908 Inverse-name of CONTROLLED meta-entity is the same as the assigned-descriptive-name of a CONTROLLED meta-entity.  
5.1.1.13
- E01909 Substitute-name of CONTROLLED meta-entity is the same as the assigned-access-name of a CONTROLLED meta-entity.  
5.1.1.13
- E01910 Substitute-name of CONTROLLED meta-entity is the same as the assigned-descriptive-name of a CONTROLLED meta-entity.  
5.1.1.13
- E01911 IRD/IRD Schema inconsistency. Entity has no corresponding type.  
5.1.1.13
- E01912 IRD/IRD Schema inconsistency. Relationship has no corresponding type.  
5.1.1.13
- E01913 IRD/IRD Schema inconsistency. Entity or Relationship has attribute with no corresponding type.  
5.1.1.13
- E01914 IRD/IRD Schema inconsistency. Entity or relationship has a group with no corresponding type.  
5.1.1.13
- E01915 IRD/IRD Schema inconsistency. Attribute has invalid format or value.  
5.1.1.13
- E01916 IRD/IRD Schema inconsistency. Attribute-group is not consistent with corresponding type.  
5.1.1.13

- E01917 IRD/IRD Schema inconsistency. Corresponding type for attribute not associated with type of corresponding entity.  
5.1.1.13
- E01918 IRD/IRD Schema inconsistency. Corresponding type for attribute-group not associated with type of corresponding entity.  
5.1.1.13
- E01919 IRD/IRD Schema inconsistency. Corresponding type for attribute not associated with type of corresponding relationship.  
5.1.1.13
- E01920 IRD/IRD Schema inconsistency. Corresponding type for attribute-group not associated with type of corresponding relationship.  
5.1.1.13
- E01921 IRD/IRD Schema inconsistency. Attribute or group repeats too many times.  
5.1.1.13
- E01922 IRD/IRD Schema inconsistency. Duplicate keys in repeating attribute-group.  
5.1.1.13
- E01923 IRD/IRD Schema inconsistency. Incomplete key in attribute-group.  
5.1.1.13
- E01924 IRD/IRD Schema inconsistency. Duplicate sequence attribute.  
5.1.1.13
- E01925 IRD/IRD Schema inconsistency. Duplicate relationship keys.  
5.1.1.13
- E01926 IRD/IRD Schema inconsistency. Entity has invalid variation-name.  
5.1.1.13
- E01927 IRD/IRD Schema inconsistency. Entity has invalid assigned-access-name.  
5.1.1.13
- E01928 IRD/IRD Schema inconsistency. Entity has invalid assigned-descriptive-name.  
5.1.1.13
- W01001 End of input encountered and last command not properly terminated.  
5
- W01002 One or both of the IRDSs are not in STANDARD mode.  
5.3.1.3
- W01004 Command executed with errors.  
5 5.2.1.6
- W01005 List specifies nonexistent entities.  
5.2.1.3
- W01006 No relationships obtained.  
5.2.1.6

- W01008 One or more relationships have not been duplicated for the new-version of the entity modified.  
5.2.1.2
- W01009 Specified entities bypassed.  
5.3.1.2
- W01012 No entities selected.  
5
- W01014 Sort command-clause ignored.  
5.2.2.1 5.2.2.2
- W01019 Incorrect panel-name specified.  
5.3.2.5
- W01020 Attribute does not exist and cannot be erased.  
6.3 6.4 6.8 6.11
- W01021 Only null-string specified. Command-clause ignored.  
6.7 6.109
- W01022 Null string specified. Line-number skipped.  
6.7 6.109
- W01023 Assigned line numbers will exceed maximum.  
6.7 6.109
- W01024 All values to be replaced.  
6.8 6.10 6.110 6.111
- W01025 Attribute is to be erased.  
6.8
- W01026 Attribute-group cannot be erased.  
6.10
- W01027 Attribute-group is to be erased.  
6.10
- W01028 Specified range contains no lines.  
6.11
- W01029 Defaulted or specified increment value has been over-ridden.  
6.29 6.125
- W01030 Duplicate relationship-type.  
6.72
- W01031 Entire text-attribute suppressed.  
6.73
- W01032 Meta-attribute does not exist and cannot be erased.  
6.106 6.107 6.110 6.112

- W01033 Meta-attribute is to be erased.  
6.110
- W01034 Meta-attribute-group cannot be erased.  
6.111
- W01035 Meta-attribute-group is to be erased.  
6.111
- W01036 Specified meta-attributes have been ignored.  
6.115
- W01037 Creation of corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity.  
5.1.1.2 5.1.1.10
- W01038 Meta-attribute-type not associated with meta-entity-type or meta-relationship-type.  
5.1.2

## 8 Panel Interface

### Introduction

The IRDS Panel Interface describes the content, not the form of an interactive, screen-oriented interface to the IRDS. The IRDS Panel Interface shall provide the same semantics as the commands which are available through the specified standard IRDS Command Language Interface.

### The Panel Concept

Since these specifications cannot assume the physical characteristics of a screen-oriented interface, it is necessary to define a "logical screen" concept. It is assumed that a logical screen can accommodate any given panel. The implementor of this interface is required to provide the semantics of each command specified. These semantics are expected to be provided in terms of a set of panels, each panel set corresponding to a command. Panel sets may overlap.

The implementor of this interface shall provide all facilities identified by the concept of panel areas. A panel area is a logical partition of a panel, and is always associated with a particular type of information or user interaction requirement.

Each panel shall have a name, unique within the set of panel names, which can be used to reference the panel. The user shall be able to use the panel's name to transfer to it. An installation shall be allowed to rename the implementor provided panel names to facilitate customization.

### Structure of the Panel Interface

The Panel Interface provides a system of panels which, when properly traversed, results in the selection and execution of any of the valid IRDS functions. Thus, this interface shall have an inherent "interpanel structure" which defines a default progression of panels displayed to the user when performing a particular IRDS function. This default progression can always be

overridden by a user by transferring control to another named panel in the Panel Interface, as long as these navigational movements are logically consistent and do not affect IRD Schema or IRD integrity.

The Panel Interface shall provide a panel, called the "Home Panel", which is the first panel encountered by a user entering the interface. Conceptually, the Home Panel is the "topmost" panel of the interface, and it is from this panel that the user is able to gain access to the entire inter-panel structure. When the HELP option is selected from the Home Panel, in addition to providing the normal help information on the use of the Home Panel, the system will provide the opportunity for a user to enter a generalized help tutorial for the Panel Interface.

The structure of the Panel Interface is discussed in terms of panel trees, panel types, and panel areas. These concepts are presented below.

### **Panel Trees**

In the specification of the Panel Interface, the concept of panel trees is used as a convenient way of organizing panels. A panel tree is defined as one or more panels which are used to represent the semantics of a single command of the IRDS Command Language Interface. Thus, there shall be a one-to-one correspondence between the set of panel trees in the Panel Interface and the set of IRDS commands. Each panel tree shall have a "root panel", which is a panel that represents the logical beginning point from which the remainder of the tree can be traversed; this root panel shall be the panel of entry from the Command Language Interface when a command name is specified in the enter-panel-dialogue-command (See Subsection 5.3.2.5). The Home Panel shall provide access to the root panel of any panel tree.

### **Panel Types**

There are two types of panels in the Panel Interface, namely IRD Schema panels and IRD panels. An IRD panel is a panel which is found within a panel tree which processes a function equivalent to one of the IRD commands found in Subsection 5.2. An IRD Schema panel is one which is found within a panel tree which processes a function equivalent to one of the IRD Schema commands described in Subsection 5.1.

The major distinction between these panel types is in the panel areas which shall be present when each panel type is displayed. This distinction can be seen below in the discussion of panel areas in the Panel Interface.

### **Panel Areas**

The panel areas which shall be present for a particular IRDS panel are dependent upon the type of panel displayed: whether it is an IRD Schema panel or an IRD panel. At least five distinct panel areas shall exist for IRD Schema panels in the Panel Interface:

- (1) The State Area.
- (2) The Data Area.
- (3) The Action Area.
- (4) The Message Area.
- (5) The Help Area.

The global characteristics of these IRD Schema panel areas are discussed in Subsection 8.1. The panel tree specifications which follow in Subsection 8.2 address those panel area characteristics which are specific to a particular IRD Schema function.

There are at least six distinct panel areas which shall exist for IRD panels in the Panel Interface. These are:

- (1) The State Area.
- (2) The Data Area.
- (3) The IRD Schema Area.
- (4) The Action Area.
- (5) The Message Area.
- (6) The Help Area.

As with the IRD Schema panel areas, the global characteristics of the IRD panel areas are discussed in Subsection 8.3. The panel tree specifications found in Subsections 8.4 and 8.5 address those panel area characteristics which are specific to a particular IRD or IRD-IRD Interface function.

### **The Operation of the Panel Interface**

/\* It is possible, and indeed likely, that the panel trees of the Panel Interface will consist of more than one panel. In other words, it is likely that the specification of IRDS commands will take place in a number of steps, each of which shall correspond to one or a group of panels. Thus, the issue arises of how to determine what panel operations are to be processed, and when these operations are to take place. This issue is addressed with the definition of the "COMMIT" function. \*/

The COMMIT function shall appear as an option in the Action Area of at least one panel in every panel tree. The selection of this function by the user shall indicate that the user wishes to process whatever IRDS functions have been specified thus far in the panel tree in which the user is currently operating. The rules associated with the operation of the COMMIT function are as follows:

- (1) The COMMIT function shall appear in the Action Area of at least one panel in every panel tree. The COMMIT function may appear in as many panels as shall be deemed necessary by the implementor.
- (2) The implementor shall provide a message to the user upon selection of the COMMIT function, indicating that a particular IRDS function has been processed.
- (3) The implementor may implement additional "implicit" COMMIT operations within the Panel Interface. That is, the implementor may choose to process certain operations at a particular point within a panel tree whether or not the user has explicitly selected the COMMIT function. These implementor-defined COMMITs shall take place only upon moving from one panel to another (i.e., they shall occur synchronously with panel transfers). As in the case of explicit COMMIT requests, the implementor shall provide a message to the user indicating that an implicit COMMIT has been performed and that a particular IRDS function has been processed.

For operations involving the deletion of entities, relationships, meta-entities, or meta-relationships, the user shall have the option of performing the deletion either with or without confirmation. When confirmation is requested, a full display of the entity/meta-entity or the relationship/meta-relationship (including all of the attributes/meta-attributes) shall be provided in the Data Area and the user shall be given the opportunity to proceed with or bypass the deletion. When confirmation is not requested, the deletion shall be performed with no such display in the Data Area.

## Special Features of the Panel Interface

Two special features shall be implemented in the Panel Interface which provide special capabilities, including:

- (1) Saving a panel.
- (2) Setting a marker on a panel.

### Saving a Panel

A user can "save" a panel which is being worked on until the next session. If this action is designated at system log-off, the last panel being worked on (and, in some cases, certain associated panels) shall be saved in their current form for retrieval at the beginning of the next IRDS session.

In the case where any of these panels contain data which was originally extracted from the IRD, this data shall not be retrieved anew by the system. It is the responsibility of the user to do this if desired.

### Setting a Marker on a Panel

At any point during a Panel Interface session, a user can choose an option which allows the current panel to be "marked". This feature shall allow the user to move to any other panel or panels that result in the display of IRD Schema or IRD information, while the marked panel remains intact and can be referenced directly at any time later in the session. After marking a panel, movement to a panel that modifies the IRD Schema or IRD shall not be allowed until the mark is removed, as any such modification can affect the integrity of the contents of the marked panel. When a user has marked a panel and subsequently attempts to EXIT the Panel Interface without returning to the marked panel:

- (1) The user shall be provided with a warning message indicating that a marked panel exists, and
- (2) The user shall be provided with the opportunity to cancel the EXIT request and return to the marked panel.

## 8.1 IRD Schema Panel Areas

**Function** To identify the panel areas which make up an IRD Schema panel in the Core standard Panel Interface.

### Format

/\* None. This breakdown is explanatory only. \*/

```

panel-area ::=
    state-area           8.1.1
    | data-area          8.1.2
    | action-area       8.1.3
    | message-area      8.1.4
    | help-area         8.1.5
    | implementor-defined-IRD-schema-panel-area

```

```
implementor-defined-IRD-schema-panel-area ::=
    /* implementor-defined-format */
```

**Syntax Rules** None.

**General Rules** None.

### 8.1.1 State Area

**Function** The State Area shall provide the following information to a user:

- (1) The name of the current panel.
- (2) The name of the IRD against which the user is operating.
- (3) The function which is currently being performed in the IRDS, for example, "Adding A Meta-Entity", or "Deleting A Meta-Relationship"; or, alternately, what the system is doing at a given point in time, for example "Updating The IRD Schema", or "Retrieving IRD Schema Information".
- (4) The name of the meta-entity being operated on, if any.
- (5) The meta-relationship-type being operated on, if any, along with the meta-entity-access-names of the meta-entities which participate in the meta-relationship.
- (6) The name of the IRD-schema-view which is currently in effect.
- (7) Other appropriate information related to the current state of the particular function being performed.

The information presented within the State Area shall be displayed either in its entirety or in segmented form as a series of sub-areas, at the option of the implementor. The contents of the State Area shall include a subset of the information generated when performing the session-status-command discussed in Subsection 5.3.2.2.

**Format** None.

**Syntax Rules** None.

### General Rules

- (1) The State Area shall always be available for display by the user.
- (2) The State Area shall be updated as changes are made to the session status.

**Security Rules** None.

### 8.1.2 Data Area

**Function** The Data Area shall display:

- (1) Labels that identify the data to be input by the user, and which show the areas to be used for data entry, or
- (2) The results of a requested output function, if the user is retrieving information.

**Format** None.

**Syntax Rules** None.

**General Rules** None.



**Security Rules** None.

### 8.1.3 Action Area

**Function** The Action Area shall indicate those actions that a user can perform for the current IRD Schema panel.

**Format** None.

**Syntax Rules** None.

### General Rules

- (1) For at least one panel within each IRD Schema panel tree, this panel area shall contain the COMMIT function:
  - (a) Selection of the COMMIT function by the user indicates a desire to process the command syntax supplied thus far (or, since the last COMMIT) in the current panel tree. The operation of the COMMIT function is discussed in detail in the discussion of the Operation of the Panel Interface in Section 8.
- (2) For every IRD Schema panel, this panel area shall contain the following functions relating to user assistance operations:
  - (a) A "HELP" function which provides the user with help information in the Help Area of the current panel. This function is equivalent to execution of the help-command described in Subsection 5.3.2.3.
  - (b) A "MARK" function which allows the current panel to be marked so that a user can transfer to another panel, yet return to the marked panel later in the session.
  - (c) A "SAVE" function which allows the user to save the current panel, along with those panels which were above it in the same panel tree and which were traversed subsequent to the last time the "COMMIT" function was specified.
  - (d) A "DEFAULTS" function, which allows the user to set or reset the session defaults displayed in the State Area (see Subsection 8.1.1). The defaults which can be set or reset include a subset of those described in the set-session-defaults-command found in Subsection 5.3.2.1.
- (3) For every IRD Schema panel, this panel area shall contain the following functions relating to panel transfer operations:
  - (a) A "NEXT PANEL" option, which shall position the user at the next panel in the default panel sequence.
  - (b) A "PREVIOUS PANEL" option, which shall position the user at the panel immediately before the current panel in the panel tree. This option shall not appear in the Home Panel.
  - (c) A "MARKED PANEL" option, which shall position the user at the last panel, if any, in which the user executed the MARK function.
  - (d) A "HOME PANEL" option, which shall position the user at the Home Panel.

- (e) A "COMMAND" option, which shall place the user in the Command Language Interface of the IRDS. This option shall not be provided in the event that no Command Language Interface is available.
  - (f) A "SELECT PANEL" option, which shall allow the user to specify the name of a panel to be transferred to.
  - (g) An "EXIT" option, which shall exit the control of the IRDS. This option shall perform a function similar to that of the exit-IRDS-system-command described in Subsection 5.3.2.4.
- (4) This panel area shall also contain options for any additional actions which can be performed within the current IRD Schema panel.

**Security Rules** None.

#### 8.1.4 Message Area

**Function** To display messages from the IRDS to the user. These include:

- (1) Messages corresponding to error and warning conditions encountered in the current IRD Schema panel.
- (2) Messages indicating successful execution of IRD Schema functions.
- (3) Other messages, as appropriate, concerning system activities which are in progress.

**Format** None.

**Syntax Rules** None.

#### General Rules

- (1) This panel area shall contain all messages which correspond to error or warning conditions which occur during the operation of the functions specified in the current IRD Schema panel.
- (2) The Message Area shall also provide messages to the user concerning notification of successful completion of any IRD Schema function.

**Security Rules** None.

#### 8.1.5 Help Area

**Function** The Help Area displays information that the system provides in response to a request for "Help" by the user. In addition, the Help Area also displays appropriate options and limitations which are in effect for a particular IRD Schema panel.

As in the case of the State Area, the information presented within the Help Area may be displayed either in its entirety or in segmented form as a series of sub-areas, at the option of the implementor.

**Format** None.

**Syntax Rules** None.

**General Rules**

- (1) While the precise format and wording of the help messages which appear in the Help Area are implementor-defined, the Help Area shall provide the following types of information:
  - (a) General overview information, which provides a general overview of the purpose and operation of the IRD Schema panel at which the user currently resides.
  - (b) Action information, which explains those actions which will take place as a result of selecting any of the options available in the Action Area of the current panel.
  - (c) Panel transfer information, which provides the user with information concerning the panel transfer options available from the current panel.
- (2) The Help Area shall also provide the user with information explaining error and warning conditions which appear in the Message Area of the current IRD Schema panel, and the actions which can be taken to overcome these conditions.
- (3) For IRD Schema panels which are requesting data input, the IRD Schema Area shall contain validation information which corresponds to the data being entered in the Data Area of a panel. This information shall be displayed at the request of the user, and may include meta-entity naming rules, validation data for meta-attributes and meta-attribute-groups, lists of the valid meta-entity-types and meta-relationship-types, or other such rules corresponding to entries to be made in the Data Area.

**Security Rules**    None.

**8.2 IRD Schema Panel Trees**

**Function**    To identify all panel trees which perform functions equivalent to a single Command Language Interface command for the IRD Schema.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

IRD-schema-panel-tree ::=
    IRD-schema-maintenance-panel-trees      8.2.1
    | IRD-schema-output-panel-trees         8.2.2
  
```

**Syntax Rules**    None.

**General Rules**    None.

**Security Rules**    None.

**Actions**        None.

**Error and Warning Conditions**    None.

**8.2.1 IRD Schema Maintenance Panel Trees**

**Function** To identify all panel trees which update the contents of the IRD Schema, and the rules which apply to all such panel trees.

**Format**

/\* None. This breakdown is explanatory only. \*/

IRD-schema-maintenance-panel-tree ::=	
add-meta-entity-panel-tree	8.2.1.1
modify-meta-entity-panel-tree	8.2.1.2
delete-meta-entity-panel-tree	8.2.1.3
add-meta-relationship-panel-tree	8.2.1.4
modify-meta-relationship-panel-tree	8.2.1.5
delete-meta-relationship-panel-tree	8.2.1.6
modify-meta-entity-access-name-panel-tree	8.2.1.7
modify-meta-entity-descriptive-name-panel-tree	8.2.1.8
modify-meta-entity-life-cycle-phase-panel-tree	8.2.1.9
copy-meta-entity-panel-tree	8.2.1.10
deactivate-IRD-panel-tree	8.2.1.11
restore-IRD-schema-panel-tree	8.2.1.12
activate-IRD-panel-tree	8.2.1.13

**Syntax Rules**

- (1) The Syntax Rules which shall exist for user-supplied information in the Data Area shall include those which exist in the specifications for the equivalent function in the Command Language Interface. The Syntax Rules which do not apply are those which relate to command-clause ordering within a particular command specification. Refer to the IRD Schema Maintenance Commands, Subsection 5.1.1, for a description of these rules.

**General Rules**

- (1) The General Rules which shall exist for these panel trees shall include those which exist for the equivalent command, with all of its associated command-clauses, in the Command Language Interface. Refer to Subsection 5.1.1 and each corresponding command description for these rules.

**Access Rules**

- (1) The Access Rules which shall exist for these panel trees shall include those which are provided for IRD Schema maintenance commands in the Command Language Interface. Refer to Access Rules Subsection 5.1.1 for a complete discussion of these rules.

**Actions**

- (1) The Actions which shall be performed by this panel tree shall be equivalent to those which are performed by the equivalent command in the Command Language Interface. Refer to Subsection 5.1.1 for a description of these actions.

## **Error and Warning Conditions**

- (1) Error and Warning Conditions which can appear in the Message Area of the panel(s) in the panel trees described in the following subsections are the same as those which exist for the equivalent IRD maintenance command in Subsection 5.1.1. Error and Warning Conditions which refer to improper ordering of command-clauses shall not apply in the Panel Interface.

### **8.2.1.1 Add Meta-Entity Panel Tree**

**Function** This panel tree shall be used to add a new meta-entity to the IRD Schema and establish initial values for specified meta-attributes and meta-attribute-groups.

#### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request and accept input of the following information:
  - (a) The meta-entity-access-name of the meta-entity being added. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to the new meta-entity.
  - (b) The corresponding meta-entity-type.
  - (c) The meta-entity-descriptive-name of the meta-entity being added. The entry of a meta-entity-descriptive-name shall be optional.
  - (d) The valid user-specified meta-attributes and meta-attribute-groups which are required or optional for meta-entities of the specified meta-entity-type.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the IRD Schema life-cycle-phase in which the new meta-entity exists.

#### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall be able to provide the following information to the user:
  - (a) Rules for meta-entity naming.
  - (b) A list of valid meta-entity-types available to the user.
  - (c) Rules for applicable meta-attributes and meta-attribute-groups, including values or ranges in an attribute-type-validation-data descriptor.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide the following options to the user:
  - (a) Add the specified meta-entity, in which case the specified meta-entity shall be added to the IRD Schema.

- (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.
- (c) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

### **8.2.1.2 Modify Meta-Entity Panel Tree**

**Function** This panel tree shall be used to modify the meta-attributes and meta-attribute-groups of an existing meta-entity in the IRD Schema.

#### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the meta-entity-access-name of an existing meta-entity. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number for the meta-entity being modified.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the following:
  - (a) The meta-entity-descriptive-name of the meta-entity being modified, if one exists.
  - (b) The system-maintained meta-attributes and meta-attribute-groups associated with the meta-entity being modified.
  - (c) The IRD Schema life-cycle-phase in which the meta-entity being modified exists.
- (3) The Data Area of the panel(s) in this panel tree shall display and allow updates to the user-specified meta-attributes and meta-attribute-groups which are or can be assigned to the meta-entity being modified.

#### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) Rules for meta-entity naming.
  - (b) Rules for applicable meta-attributes and meta-attribute-groups.
  - (c) Rules which ensure that the integrity of the IRD shall not be compromised. Refer to the General Rules of Subsection 5.1.1.2.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Modify the specified meta-entity, in which case the specified meta-entity shall be modified.
  - (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.

- (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before modifications were entered by the user.

### 8.2.1.3 Delete Meta-Entity Panel Tree

**Function** This panel tree shall be used to delete an existing meta-entity from the IRD Schema.

#### Data Area Contents

- (1) The Data Area of the panel(s) in this panel tree shall request input of the meta-entity-access-name of an existing meta-entity in the IRD Schema. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number for the meta-entity being deleted.
- (2) Optionally, the Data Area of the panel(s) in this panel tree shall display the meta-entity-descriptive-name, the meta-attributes and meta-attribute-groups, and the IRD Schema life-cycle-phase associated with the meta-entity being deleted. This information shall be displayed only when the user selects the Delete with Confirmation option in the Action Area.

#### Help Area Contents

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) Rules for meta-entity naming.
  - (b) General rules which may prevent the meta-entity from being deleted, including those which ensure that the integrity of the IRD shall not be compromised. Refer to General Rules of Subsection 5.1.1.

#### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide the following options to the user:
  - (a) Delete with confirmation, in which case the specified meta-entity's meta-entity-access-name, associated meta-attributes and meta-attribute-groups, and IRD Schema life-cycle-phase, are displayed in the Data Area. The user is then provided the opportunity to proceed with or bypass the deletion.
  - (b) Delete the specified meta-entity, in which case the specified meta-entity shall be deleted from the IRD Schema without confirmation.
  - (c) Display rules, in which case the rules which may prohibit the deletion of the meta-entity shall appear in the Help Area.

### 8.2.1.4 Add Meta-Relationship Panel Tree

**Function** This panel tree shall be used to establish a meta-relationship between two meta-entities in the IRD Schema and to establish initial values for specified meta-attributes and meta-attribute-groups.

### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request and accept input of the following information:
  - (a) The meta-entity-access-names of the meta-entities being related. If full version-identifiers for the meta-entities are not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to these meta-entities.
  - (b) The meta-relationship-class-type for the new meta-relationship.
  - (c) The sequencing meta-attribute of the meta-relationship, for a meta-relationship-type which is sequenced.
  - (d) The user specifiable meta-attributes and meta-attribute-groups which can be assigned to the meta-relationship being added.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the IRD Schema life-cycle-phase in which each of the two participating meta-entities exist.

### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) A list of valid meta-relationship-types which are available to the user.
  - (b) Rules for meta-entity naming.
  - (c) A list of valid meta-entity-types.
  - (d) For each meta-relationship-type, the identification of the applicable sequencing meta-attribute-type, if any.
  - (e) Rules for applicable meta-attributes and meta-attribute-groups.

### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Add the specified meta-relationship, in which case the specified meta-relationship shall be added to the IRD Schema.
  - (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.
  - (c) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.



### 8.2.1.5 Modify Meta-Relationship Panel Tree

**Function** This panel tree shall be used to modify the user-specified meta-attributes and meta-attribute-groups of an existing meta-relationship in the IRD Schema.

#### Data Area Contents

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) The meta-entity-access-names of two existing meta-entities in the IRD Schema. If full version-identifiers are not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to each of the meta-entities.
  - (b) The meta-relationship-class-type of the meta-relationship.
  - (c) The sequencing meta-attribute of the meta-relationship, for a meta-relationship-type which is sequenced.
- (2) The Data Area of the panel(s) in this panel tree shall display and allow updates to all user-specified meta-attributes and meta-attribute-groups which are or can be associated with the meta-relationship being modified.
- (3) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to:
  - (a) The IRD Schema life-cycle-phase in which each of the specified meta-entities exist.
  - (b) The system-maintained meta-attributes and meta-attribute-groups associated with the meta-relationship being modified.

#### Help Area Contents

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) A list of valid meta-relationship-types which are available to the user.
  - (b) Rules for meta-entity naming.
  - (c) A list of valid meta-entity-types.
  - (d) For each meta-relationship-type, the identification of the applicable sequencing meta-attribute-type, if any.
  - (e) Rules for applicable meta-attributes and meta-attribute-groups.
  - (f) General rules which may prevent the meta-relationship from being modified, including those which ensure that the integrity of the IRD shall not be compromised. Refer to General Rules Subsection 5.1.1.

### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Modify the specified meta-relationship, in which case the specified modifications are made to the meta-attributes and meta-attribute-groups of the meta-relationship.
  - (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.
  - (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before modifications were entered by the user.

#### **8.2.1.6 Delete Meta-Relationship Panel Tree**

**Function** This panel tree shall be used to delete an existing meta-relationship from the IRD Schema.

### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) The meta-entity-access-names of the two meta-entities in the IRD Schema which participate in the meta-relationship to be deleted. If full version-identifiers are not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to each of the meta-entities.
  - (b) The meta-relationship-class-type of the meta-relationship to be deleted.
  - (c) The sequencing meta-attribute of the meta-relationship, for a meta-relationship-type which is sequenced.
- (2) Optionally, the Data Area of the panel(s) in this panel tree shall display the meta-entity-descriptive-names of the two meta-entities which participate in the specified meta-relationship, the applicable meta-relationship-class-type, the IRD Schema life-cycle-phase in which each of the meta-entities exist, and the meta-attributes and meta-attribute-groups which are associated with the meta-relationship being deleted. This information shall be displayed only when the user selects the Delete with Confirmation option in the Action Area.

### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) A list of valid meta-relationship-types which are available to the user.
  - (b) Rules for meta-entity naming.
  - (c) A list of valid meta-entity-types.

- (d) For each meta-relationship-type, the identification of the applicable sequencing meta-attribute-type, if any.
- (e) Rules for the sequencing meta-attribute-type.
- (f) General rules which may prevent the meta-relationship from being deleted, including those which ensure that the integrity of the IRD shall not be compromised. Refer to General Rules of Subsection 5.1.1.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Delete with confirmation, in which case the two meta-entity-descriptive-names which participate in the specified meta-relationship, the applicable meta-relationship-class-type, the IRD Schema life-cycle-phases in which the meta-entities exist, and the associated meta-attributes and meta-attribute-groups, are displayed in the Data Area. The user is then provided the opportunity to proceed with or bypass the deletion.
  - (b) Delete the specified meta-relationship, in which case the specified meta-relationship shall be deleted from the IRD Schema without confirmation.
  - (c) Display validation rules, in which case the desired validation information shall appear in the Help Area.

#### **8.2.1.7 Modify Meta-Entity-Access-Name Panel Tree**

**Function** This panel tree shall be used to change the meta-entity assigned-access-name of all meta-entities with a given assigned-access-name.

#### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) The meta-entity assigned-access-name of at least one existing meta-entity.
  - (b) A new meta-entity assigned-access-name, which shall be neither the assigned-access-name nor the assigned-descriptive-name of an existing meta-entity.

#### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) Rules for meta-entity naming.
  - (b) Rules for this command.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:

- (a) Modify the assigned-access-name, in which case each meta-entity with the specified assigned-access-name shall be changed to have an assigned-access-name equal to the new meta-entity-assigned-access-name. Version-identifiers remain unchanged.
- (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.
- (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before user input.

#### **8.2.1.8 Modify Meta-Entity-Descriptive-Name Panel Tree**

**Function** This panel tree shall be used to change the meta-entity assigned-descriptive-name of all meta-entities with a given assigned-descriptive-name.

##### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) The meta-entity assigned-descriptive-name of at least one existing meta-entity.
  - (b) A new meta-entity assigned-descriptive-name, which shall not be the assigned-descriptive-name of an existing meta-entity. The new meta-entity assigned-descriptive-name shall also not be equal to the assigned-access-name of any meta-entity in the IRD Schema except those meta-entities identified by the specified meta-entity assigned-descriptive-name.

##### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) Rules for meta-entity naming.
  - (b) Rules for this command.

##### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Modify the meta-entity assigned-descriptive-name, in which case each meta-entity with the specified meta-entity assigned-descriptive-name shall be changed to have an assigned-descriptive-name equal to the new meta-entity assigned-descriptive-name. Version-identifiers remain unchanged.
  - (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.
  - (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before user input.

**8.2.1.9 Modify Meta-Entity Life-Cycle-Phase Panel Tree**

**Function** This panel tree shall be used to move one or more meta-entities from one IRD Schema life-cycle-phase to another.

**Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) One or more meta-entity-access-names. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to each meta-entity specified.
  - (b) The current IRD Schema life-cycle-phase in which the given meta-entity(s) exists.
  - (c) The IRD Schema life-cycle-phase into which the specified meta-entity(s) shall be moved.

**Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide:
  - (a) Rules for meta-entity naming.
  - (b) The names of existing IRD Schema life-cycle-phases.
  - (c) Rules for this command.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide the following options:
  - (a) Move the specified meta-entity(s) into the new IRD Schema life-cycle-phase.
  - (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.

**8.2.1.10 Copy Meta-Entity Panel Tree**

**Function** This panel tree shall be used to create a meta-entity with the same user-specifiable meta-attributes and meta-attribute-groups (and, optionally, the meta-relationships to the same meta-entities) as an existing meta-entity.

**Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) The meta-entity-access-name of an existing meta-entity. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number for the specified meta-entity.

- (b) The meta-entity-access-name of a new meta-entity, which may include a new version identifier for the existing meta-entity. If the user enters a meta-entity assigned-access-name only, then the system shall provide and display the default variation-name and revision-number assigned to the new meta-entity.
  - (c) A meta-entity-descriptive-name for the new meta-entity. Entry of this information shall be optional.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the following information:
- (a) The IRD Schema life-cycle-phase in which each of the specified meta-entities exist.
  - (b) The meta-attributes and meta-attribute-groups associated with the meta-entity being copied.

#### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide:
- (a) Rules for meta-entity naming.
  - (b) Rules for this command.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide the following options:
- (a) To copy the specified meta-entity without meta-relationships, in which case the specified existing meta-entity shall be copied to the new specified entity, without copying the meta-relationships in which the existing meta-entity participates.
  - (b) To copy the specified meta-entity with meta-relationships, in which case the specified existing meta-entity shall be copied to the specified new meta-entity, along with all meta-relationships in which the existing meta-entity participates.
  - (c) Display validation rules, in which case the desired validation information shall appear in the Help Area.
  - (d) Refresh the Data Area, in which case all user-specified data shall be erased from the Data Area.

#### **8.2.1.11 Deactivate IRD Panel Tree**

**Function** This panel tree shall be used to stop all IRD activity and restrict access to the IRD Schema to a single user.

**Data Area Contents** None.

**Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide rules for this command.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide the following options:
  - (a) Deactivate all IRD activity. The commands which are disabled by this action are specified in Actions of Subsection 5.1.1.
  - (b) Display command rules, in which case the desired information shall appear in the Help Area.

**8.2.1.12 Restore IRD Schema Panel Tree**

**Function** This panel tree shall be used to restore the IRD Schema to its state as of the last time a deactivate-IRD action was issued.

**Data Area Contents** None.

**Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide rules for this command.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide the following options:
  - (a) Restore the IRD Schema, in which case the IRD Schema shall be restored to its state as of the execution of the last deactivate-IRD action.
  - (b) Display command rules, in which case the desired information shall appear in the Help Area.

**8.2.1.13 Activate IRD Panel Tree**

**Function** This panel tree shall be used to enable IRD activity after the IRD has been deactivated.

**Data Area Contents** None.

**Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide rules for this command.

### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide the following options:
  - (a) Activate IRD. The commands which are enabled by this action are specified in Actions of Subsection 5.1.1.
  - (b) Display command rules, in which case the desired information shall appear in the Help Area.

### 8.2.2 IRD Schema Output Panel Trees

**Function** To identify the panel trees used to output the contents of the IRD Schema.

#### Format

/\* None. This breakdown is explanatory only. \*/

```
IRD-schema-output-panel-tree ::=
    IRD-schema-output-panel-tree           8.2.2.1
    | inquire-meta-entity-panel-tree       8.2.2.2
    | inquire-meta-relationship-panel-tree  8.2.2.3
```

#### Syntax Rules

- (1) The Syntax Rules which shall exist for user-supplied information in the Data Area shall include those which exist in the specifications for the equivalent functions in the Command Language Interface, except that the Syntax Rules which relate to command-clause ordering within a particular command specification shall not apply. Refer to the IRD Schema Output Command, Subsection 5.1.2, for a description of these rules. The inquire-meta-entity and inquire-meta-relationship panel trees shall not allow user modification of IRD Schema information.

#### General Rules

- (1) The General Rules which shall exist for this panel tree shall include those which exist for the equivalent command, with all of its associated command-clauses, in the Command Language Interface. Refer to Subsection 5.1.2 for a description of these rules.

**Security Rules** None.

#### Actions

- (1) The Actions which shall be performed by this panel tree shall be equivalent to those which are performed by the equivalent command in the Command Language Interface. Refer to Subsection 5.1.2.

#### Error and Warning Conditions

- (1) Error and Warning Conditions which can appear in the Message Area of one or more of the panels in the panel trees described in the following Subsections are the same as those which exist for the equivalent IRD Schema output command found in Subsection 5.1.2.



### 8.2.2.1 IRD Schema Output Panel Tree

**Function** This panel tree shall be used to retrieve IRD Schema descriptors, format output, and route the output to a specified destination. The functionality of this panel tree shall be equivalent to that found in the IRD-schema-output-command, Subsection 5.1.2.

#### Data Area Contents

- (1) The Data Area of the panel(s) in this panel tree shall request and accept input of the following information:
  - (a) The selection criteria for the meta-entities to be included in the output.
  - (b) The sort parameters for sorting the set of meta-entities into the order they are to appear in the output. The entry of this information shall be optional.
  - (c) The show options, which determine what IRD Schema information shall be displayed in the output for the selected meta-entities.
  - (d) The destination to which the output shall be routed. The entry of this information shall be optional. If not entered, an implementation-specified default destination shall be used.

#### Help Area Contents

- (1) In addition to the contents identified in Subsection 8.1.5, the Help Area of the panel(s) in this tree shall provide the following information to the user:
  - (a) Rules for specifying selection criteria.
  - (b) The valid show options.
  - (c) Rules for sort parameters.
  - (d) Valid destinations.

#### Action Area Contents

- (1) In addition to those global options specified in General Rules of Subsection 8.1.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Display the specified output. The specified output shall be routed to the selected destination.
  - (b) Display rules, in which case the rules for selection criteria and/or sort parameters shall appear in the Help Area.
  - (c) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

### **8.2.2.2 Inquire Meta-Entity Panel Tree**

**Function** This panel tree shall be used to display an existing meta-entity as the modify-meta-entity panel tree displays it, disallowing updates.

#### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the meta-entity-access-name of an existing meta-entity. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number for the meta-entity.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the IRD Schema life-cycle-phase of the meta-entity, and all meta-attributes and meta-attribute-groups which are or can be assigned to the meta-entity being displayed.

#### **Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the rules for meta-entity naming.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Display the specified meta-entity, in which case the specified meta-entity shall be displayed.
  - (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.

### **8.2.2.3 Inquire Meta-Relationship Panel Tree**

**Function** This panel tree shall be used to display a meta-relationship as the modify-meta-relationship panel tree displays it, disallowing updates.

#### **Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) The meta-entity-access-names of two existing meta-entities in the IRD Schema. If full version-identifiers are not specified by the user, then the system shall provide and display the default variation-name and revision-number for the meta-entity.
  - (b) The meta-relationship-class-type of the existing meta-relationship.
  - (c) The sequencing meta-attribute of the meta-relationship, for a meta-relationship-type which is sequenced.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the IRD Schema life-cycle-phase of the specified meta-entities, and all meta-attributes

and meta-attribute-groups which are or can be associated with the meta-relationship being displayed.

**Help Area Contents**

- (1) In addition to what was specified in Subsection 8.1.5, the Help Area of the panel(s) in this panel tree shall provide the following information to the user:
  - (a) Rules for meta-entity naming.
  - (b) The allowed meta-relationship-types, along with the corresponding meta-relationship-class-types.
  - (c) The sequencing meta-attribute-type, if any, associated with a given meta-relationship-type.
  - (d) Validation rules for the sequencing meta-attribute-type.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.1.3, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Display the specified meta-relationship, in which case the meta-attributes and meta-attribute-groups of the specified meta-relationship are displayed.
  - (b) Display validation rules, in which case the desired validation information shall appear in the Help Area.

**8.3 IRD Panel Areas**

**Function** To identify the panel areas which make up an IRD panel in the Panel Interface.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

panel-area ::=
    state-area           8.3.1
    | data-area          8.3.2
    | IRD-schema-area   8.3.3
    | action-area       8.3.4
    | message-area      8.3.5
    | help-area         8.3.6
    | implementor-defined-IRD-panel-area

implementor-defined-IRD-panel-area ::=
    /* implementor-defined-format */
    
```

**Syntax Rules** None.

**General Rules** None.

### 8.3.1 State Area

**Function** The State Area shall provide the following information to a user:

- (1) The name of the current panel.
- (2) The name of the IRD against which the user is operating.
- (3) The function which is currently being performed in the IRDS, for example, "Adding A Record", "Deleting An Element", or "Creating An Entity List"; or, alternately, what the system is doing at a given point in time, for example "Updating The IRD", or "Retrieving Information".
- (4) The name of the entity being operated on, if any.
- (5) The relationship-type or relationship-class-type being operated on, if any, along with the entity-access-names of the entities which participate in the relationship.
- (6) The IRD-View in which the user is operating.
- (7) The Encode/Decode option which shall be in effect for attributes which are displayed.
- (8) Other appropriate information related to the current state of the particular function being performed.

The information presented within the State Area may be displayed either in its entirety or in segmented form as a series of sub-areas, at the option of the implementor. The contents of the State Area shall include the information generated when performing the session-status-command discussed in Subsection 5.3.2.2.

**Format** None.

**Syntax Rules** None.

#### General Rules

- (1) The State Area shall always be available for display by the user.
- (2) The State Area shall be updated as changes are made to the session status.

**Security Rules** None.

### 8.3.2 Data Area

**Function** To display:

- (1) Labels that identify the data to be input by the user, and which show the areas to be used for data entry, or
- (2) The results of a requested output function, if the user is retrieving information.

**Format** None.

**Syntax Rules** None.

**General Rules** None.

**Security Rules** None.

### 8.3.3 IRD Schema Area

**Function** To display appropriate IRD Schema options which are available or IRD-schema-related limitations which are in effect for a particular IRD panel.

**Format** None.

**Syntax Rules** None.

#### General Rules

- (1) For IRD panels which are requesting data input, the IRD Schema Area shall contain IRD Schema information which corresponds to the data being entered in the Data Area of a panel. This IRD Schema information shall be displayed at the request of the user, and may include entity naming rules, validation data for attributes and attribute-groups, a list of the valid IRD Schema descriptors, or other such rules contained in the IRD Schema corresponding to entries to be made in the Data Area.

**Security Rules** None.

### 8.3.4 Action Area

**Function** The Action Area shall indicate those actions that a user can perform for the current IRD panel.

**Format** None.

**Syntax Rules** None.

#### General Rules

- (1) For at least one panel within each IRD panel tree, this panel area shall contain the COMMIT function:
  - (a) Selection of the COMMIT function by the user indicates a desire to process the command syntax supplied thus far (or, since the last COMMIT) in the current panel tree. The operation of the COMMIT function is discussed in detail in the Operation of the Panel Interface in Section 8.
- (2) For every IRD panel, this panel area shall contain the following functions relating to user assistance operations:
  - (a) A "HELP" function which provides the user with help information in the Help Area of the current panel. This function shall be equivalent to execution of the help-command described in Subsection 5.3.2.3.
  - (b) A "MARK" function which allows the current panel to be marked so that a user can transfer to another panel, yet return to the marked panel later in the session.
  - (c) A "SAVE" function which allows the user to save the current panel, along with those panels which were above it in the same panel tree and which were traversed subsequent to the last time the "COMMIT" function was specified.

- (d) A "DEFAULTS" function, which allows the user to set or reset the session defaults displayed in the State Area (see Subsection 8.3.1). The defaults which can be set or reset include those described in the set-session-defaults-command found in Subsection 5.3.2.1.
- (3) For every IRD panel, this panel area shall contain the following functions relating to panel transfer operations:
- (a) A "NEXT PANEL" option, which shall position the user at the next panel in the default panel sequence.
  - (b) A "PREVIOUS PANEL" option, which shall position the user at the panel immediately before the current panel in the panel tree. This option shall not appear in the Home Panel.
  - (c) A "MARKED PANEL" option, which shall position the user at the last panel, if any, in which the user executed the mark function.
  - (d) A "HOME PANEL" option, which shall position the user at the Home Panel.
  - (e) A "COMMAND" option, which shall place the user in the Command Language Interface of the IRDS. This option shall not be provided in the event that no Command Language Interface is available.
  - (f) A "SELECT PANEL" option, which shall allow the user to specify the name of a panel to be transferred to.
  - (g) An "EXIT" option, which shall exit the control of the IRDS. This option shall perform a function similar to that of the exit-IRDS-system-command described in Subsection 5.3.2.4.
- (4) This panel area shall also contain options for any additional actions which can be performed within the current IRD panel.

**Security Rules** None.

### 8.3.5 Message Area

**Function** To display messages from the IRDS to the user. These include:

- (1) Messages corresponding to error and warning conditions encountered in the current IRD panel.
- (2) Messages indicating successful execution of IRD functions.
- (3) Other messages, as appropriate, concerning system activities which are in progress.

**Format** None.

**Syntax Rules** None.

### General Rules

- (1) This panel area shall contain all messages which correspond to error or warning conditions which occur during the operation of the functions specified in the current IRD panel.

- (2) The Message Area shall also provide messages to the user concerning notification of successful completion of any IRD function.

**Security Rules** None.

### 8.3.6 Help Area

**Function** The Help Area shall display information that the system provides in response to a request for "Help" by the user.

As in the case of the State Area, the information presented within the Help Area shall be displayed either in its entirety or in segmented form as a series of sub-areas, at the option of the implementor. The contents of the Help Area shall include the information generated when performing the help-command discussed in Subsection 5.3.2.3.

**Format** None.

**Syntax Rules** None.

#### General Rules

- (1) While the precise format and wording of the help messages which appear in the Help Area are implementor-defined, the Help Area shall provide the following types of information:
- (a) General overview information, which provides a general overview of the purpose and operation of the IRD panel at which the user currently resides.
  - (b) Action information, which explains those actions which shall take place as a result of selecting any of the options available in the Action Area of the current panel.
  - (c) Panel transfer information, which provides the user with information concerning the panel transfer options available from the current panel.
- (2) The Help Area shall also provide the user with information explaining error and warning conditions which appear in the Message Area of the current IRD panel, and the actions which can be taken to overcome these conditions.

**Security Rules** None.

### 8.4 IRD Panel Trees

**Function** To identify all panel trees which perform functions equivalent to a single Command Language Interface command.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

panel-tree ::=
    IRD-maintenance-panel-trees           8.4.1
    | IRD-output-panel-trees              8.4.2
    
```

**Syntax Rules** None.

**General Rules** None.

**Security Rules**

- (1) In order to operate against the IRD using the Panel Interface, the same conditions as those for the execution of IRD commands shall hold. These conditions are found in the Security Rules in Subsection 5.2.

**Actions** None.

**Error and Warning Conditions** None.

**8.4.1 IRD Maintenance Panel Trees**

**Function** To identify all panel trees which update the contents of the IRD, and the rules which apply to all such panel trees.

**Format**

/\* None. This breakdown is explanatory only. \*/

IRD-maintenance-panel-tree ::=	
add-entity-panel-tree	8.4.1.1
modify-entity-panel-tree	8.4.1.2
delete-entity-panel-tree	8.4.1.3
add-relationship-panel-tree	8.4.1.4
modify-relationship-panel-tree	8.4.1.5
delete-relationship-panel-tree	8.4.1.6
modify-entity-access-name-panel-tree	8.4.1.7
modify-entity-descriptive-name-panel-tree	8.4.1.8
modify-entity-life-cycle-phase-panel-tree	8.4.1.9
copy-entity-panel-tree	8.4.1.10

**Syntax Rules**

- (1) The Syntax Rules which shall exist for user-supplied information in the Data Area shall include those which exist in the specifications for the equivalent functions in the Command Language Interface, except that the Syntax Rules which relate to command-clause ordering within a particular command specification shall not apply. Refer to the IRD Maintenance Commands, Subsection 5.2.1, for a description of these rules.

**General Rules**

- (1) The General Rules which shall exist for these panel trees shall include those which exist for the equivalent command, with all of its associated command-clauses, in the Command Language Interface. Refer to Subsection 5.2.1 for a description of these rules.

**Security Rules** None.

**Actions**

- (1) The Actions which shall be performed by this panel tree shall be equivalent to those which are performed by the equivalent command in the Command Language Interface. Refer to Subsection 5.2.1. for a description of these actions.



## **Error and Warning Conditions**

- (1) Error and Warning Conditions which can appear in the Message Area of one or more of the panels in the panel trees described in the following Subsections are the same as those which exist for the equivalent IRD found in Subsection 5.2.1. Error and Warning Conditions which refer to improper ordering of command-clauses shall not apply in the Panel Interface.

### **8.4.1.1 Add Entity Panel Tree**

**Function** This panel tree shall be used to add a new entity to the IRD and establish initial values for specified attributes and attribute-groups.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The entity-access-name of the entity being added, with the option of a null-mark in the case where the new entity shall be of a type that has system-generated-names. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to the new entity.
  - (b) The entity-type of the entity being added.
  - (c) The entity-descriptive-name of the entity being added. The entry of an entity-descriptive-name shall be optional.
  - (d) A quality-indicator to be associated with the entity being added. The entry of a quality-indicator shall be optional.
  - (e) The valid attributes and attribute-groups which can be assigned to the entity being added. The entry of these attributes and attribute-groups shall be optional.
- (2) The Data Area of one or more panels in this panel tree shall display, but shall not allow updates to, the IRD-partition in which the entity being added exists.

#### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid entity-types available to the user.
  - (c) A list of valid quality-indicators.
  - (d) A list of variation-names for the entity-type being added.
  - (e) Rules for applicable attributes and attribute-groups, including values or ranges in an attribute-type-validation-data descriptor.

### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Add the specified entity, in which case the specified entity shall be added to the IRD.
  - (b) Display validation rules, in which case the desired validation information shall appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

#### **8.4.1.2 Modify Entity Panel Tree**

**Function** This panel tree shall be used to modify the attributes and attribute-groups of an existing entity in the IRD, or to create a new entity with the same assigned-access-name and different version-identifier as an existing entity.

### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
  - (a) The entity-access-name of an existing entity in the IRD. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number of the entity.
  - (b) The entity-descriptive-name of the entity being modified, if one has not already been specified. The entry of an entity-descriptive-name shall be optional.
- (2) The Data Area of one or more panels in this panel tree shall display and allow updates to the following information:
  - (a) The version-identifier of the existing entity.
  - (b) The quality-indicator associated with the entity being modified, if one exists.
  - (c) The user-specifiable attributes and attribute-groups which are or can be assigned to the entity being modified.
- (3) The Data Area of one or more panels in this panel tree shall display, but shall not allow updates to, the following information:
  - (a) The entity-type of the entity being modified.
  - (b) The IRD-partition of the entity being modified. If a new entity shall be created as a result of this panel tree, the IRD life-cycle-phase of the new entity shall also be displayed.
  - (c) The entity-descriptive-name of the entity being modified, if one exists. Modification of the entity-descriptive-name within this panel tree shall not be allowed, however.

- (d) The system-maintained meta-attributes and meta-attribute-groups associated with the entity being modified.

#### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid quality-indicators.
  - (c) Rules for applicable attributes and attribute-groups, including values or ranges in an attribute-type-validation-data descriptor.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Modify the specified entity, in which case the specified entity shall be modified or the modifications shall be made to the new entity being added to the IRD.
  - (b) Display validation rules, in which case the desired validation information shall appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before modifications were entered by the user.

#### **8.4.1.3 Delete Entity Panel Tree**

**Function** This panel tree shall be used to delete one or more existing entities from the IRD.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
  - (a) The entity-access-name of an existing entity in the IRD. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number of the entity.
- (2) The Data Area of one or more panels in this panel tree shall optionally display, but shall not allow updates to, the following information for each entity being deleted. This information shall only be displayed if the delete with confirmation or the loop with confirmation option is selected in the Action Area:
  - (a) The entity-access-name of each entity being deleted. Even if the user has entered the entity-access-name previously, the Data Area shall redisplay the entity-access-name to allow name confirmation.
  - (b) The entity-descriptive-name of each entity being deleted, if one exists.

- (c) The IRD-partition in which each entity exists.
- (d) The quality-indicator associated with each entity being deleted, if one exists.
- (e) The attributes and attribute-groups associated with each entity being deleted.

#### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Delete with Confirmation, in which case the entity-access-name, the entity-descriptive-name, the IRD-partition, the quality-indicator, and the attributes and attribute-groups associated with the specified entity are displayed in the Data Area and the user shall be given the opportunity to proceed with or bypass the entity deletion.
  - (b) Delete, in which case the entity with the specified entity-access-name is deleted from the IRD without confirmation.
  - (c) Display validation rules, in which case the desired validation information shall appear in the IRD Schema Area.

#### **8.4.1.4 Add Relationship Panel Tree**

**Function** This panel tree shall be used to establish a relationship between two entities in the IRD and to establish initial values for specified attributes and attribute-groups.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The entity-access-names of the entities being related. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to the entity.
  - (b) In the case where the second entity of the relationship does not currently exist in the IRD, the entity-type of the new entity being created.
  - (c) The relationship-type or relationship-class-type of the relationship being added.
  - (d) The order or sequence of the relationship, for a relationship-type which shall be sequenced.
  - (e) The attributes and attribute-groups to be assigned to the relationship being added. The entry of these attributes and attribute-groups shall be optional.

- (2) The Data Area of one or more panels in this panel tree shall display, but shall not allow updates to, the IRD-partition in which each of the specified entities exist.

#### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
- (a) A list of valid relationship-types which are available to the user.
  - (b) Rules for entity naming.
  - (c) A list of valid entity-types.
  - (d) Rules for applicable attributes and attribute-groups, including values or ranges in an attribute-type-validation-data descriptor.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
- (a) Add the specified relationship, in which case the specified relationship (and, if applicable, the specified new entity) shall be added to the IRD.
  - (b) Display validation rules, in which case the desired validation information shall appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

#### **8.4.1.5 Modify Relationship Panel Tree**

**Function** This panel tree shall be used to modify the attributes and attribute-groups of an existing relationship in the IRD.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
- (a) The entity-access-names of two existing entities in the IRD. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number for each of the entities.
  - (b) The name of a relationship-type or relationship-class-type.
- (2) The Data Area of one or more panels in this panel tree shall display and allow updates to all user-specifiable attributes and attribute-groups which are or can be associated with the relationship being modified.
- (3) The Data Area of one or more panels in this panel tree shall display, but shall not allow updates to:

- (a) The IRD-partition in which each of the specified entities exist.
- (b) The system-maintained attributes and attribute-groups associated with the relationship being modified.

#### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid relationship-types and/or relationship-class-types.
  - (c) Rules for applicable attributes and attribute-groups, including values or ranges in an attribute-type-validation-data descriptor.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Modify the specified relationship, in which case the specified modifications are made to the attributes and attribute-groups of the relationship.
  - (b) Display validation rules, in which case the desired validation information shall appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before modifications were entered by the user.

#### **8.4.1.6 Delete Relationship Panel Tree**

**Function** This panel tree shall be used to delete an existing relationship from the IRD.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
  - (a) The entity-access-names of the two entities in the IRD which participate in the relationship to be deleted. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number of the entities.
  - (b) The name of the relationship-type or relationship-class-type of the relationship to be deleted.
- (2) Optionally, the Data Area of one or more panels in this panel tree shall display the following information. This information shall be displayed if the user selects the delete with confirmation option in the Action Area:

- (a) The entity-access-names of the two entities which participate in the relationship to be deleted, along with the name of the relationship-type or relationship-class-type of the relationship. Although this information was previously entered by the user, it shall be redisplayed to allow confirmation.
- (b) The IRD-partition in which each of the meta-entities exist.
- (c) The attributes and attribute-groups associated with the relationship being deleted.

#### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid relationship-types and/or relationship-class-types.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following option to the user:
  - (a) Delete with Confirmation, in which case the entity-access-names which participate in the relationship, the relationship-type or relationship-class-type, the IRD-partitions in which the entities exist, and the attributes and attribute-groups associated with the specified relationship are displayed in the Data Area and the user shall be given the opportunity to proceed with or bypass the relationship deletion.
  - (b) Delete, in which case the specified relationship shall be deleted from the IRD without confirmation.
  - (c) Display validation rules, in which case the desired validation information shall appear in the IRD Schema Area.

#### **8.4.1.7 Modify Entity-Access-Name Panel Tree**

**Function** This panel tree shall be used to change the assigned-access-name of all entities with a given assigned-access-name.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
  - (a) The assigned-access-name of at least one existing entity in the IRD.
  - (b) A new assigned-access-name, which is neither the assigned-access-name nor the assigned-descriptive-name of an existing entity.
  - (c) The names of one or more IRD life-cycle-phases. Entities with the specified assigned-access-name which are in a specified IRD life-cycle-phase shall not have their assigned-access-names changed. Entry of IRD life-cycle-phase(s) shall be optional.

### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide rules for entity naming.

### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Modify the assigned-access-name, in which case each entity with the specified assigned-access-name shall be changed to have an assigned-access-name equal to the new assigned-access-name. Version-identifiers remain unchanged.
  - (b) Display rules for entity naming, in which case the desired information shall appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before user input.

#### **8.4.1.8 Modify Entity-Descriptive-Name Panel Tree**

**Function** This panel tree shall be used to change the assigned-descriptive-name of all entities with a given assigned-descriptive-name.

### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
  - (a) The assigned-descriptive-name of at least one existing entity in the IRD.
  - (b) A new assigned-descriptive-name which shall not be the assigned-descriptive-name of an existing entity. The new assigned-descriptive-name shall also not be equal to the assigned-access-name any entity in the IRD except for those entities identified by the specified assigned-descriptive-name.
  - (c) The names of one or more IRD life-cycle-phases. Entities with the specified assigned-descriptive-name which are in a specified IRD life-cycle-phase shall not have their assigned-descriptive-names changed. Entry of IRD life-cycle-phase(s) shall be optional.

### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide rules for entity naming.

### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:



- (a) Modify the assigned-descriptive-name, in which case each entity with the specified assigned-descriptive-name shall be changed to have an assigned-descriptive-name equal to the new assigned-descriptive-name. Version-identifiers remain unchanged.
- (b) Display rules for entity naming, in which case the desired information shall appear in the IRD Schema Area.
- (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before user input.

#### **8.4.1.9 Modify Entity Life-Cycle-Phase Panel Tree**

**Function** This panel tree shall be used to transfer one or more entities from one IRD Life-Cycle-Phase to another.

##### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
  - (a) The entity-access-name of an existing entity in the IRD. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number of the entity.
  - (b) The name of a valid IRD life-cycle-phase in which the specified entity currently resides.
  - (c) The name of a valid IRD life-cycle-phase, distinct from the current IRD life-cycle-phase, into which the specified entity shall be transferred.

##### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid IRD life-cycle-phase names.

##### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Transfer the specified entity from one IRD life-cycle-phase to another, in which case the specified entity shall be transferred to the specified new IRD life-cycle-phase.
  - (b) Display rules, in which case the desired entity naming rules shall appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before user input.

#### **8.4.1.10 Copy Entity Panel Tree**

**Function** This panel tree shall be used to create a new entity with the same attributes and attribute-groups (and, optionally, the relationships to the same entities) as an existing entity in the IRD.

##### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request input of the following information:
  - (a) The entity-access-name of an existing entity in the IRD. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number of the entity.
  - (b) The entity-access-name of a new entity, which may include a new version-identifier for the existing entity. If only an assigned-access-name is entered, the system shall provide and display the default variation-name and revision-number for the new entity.
  - (c) The entity-descriptive-name of the new entity. Entry of this information shall be optional, and an entity-descriptive-name shall not be entered if the entity-access-name of the new entity differs from that of the existing entity only by the version-identifier.
  - (d) A quality-indicator for the entity being copied to.
- (2) The Data Area of one or more panels in this panel tree shall display, but shall not allow updates to, the following information:
  - (a) The IRD-partition in which each of the specified entities exist.
  - (b) The attributes and attribute-groups assigned to the entity being copied.

##### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid quality-indicators.
  - (c) A list of valid variation-names for the entity-type being copied.

##### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Copy the specified entity without relationships, in which case the specified existing entity shall be copied to the specified new entity, without copying the relationships in which the existing entity participates.

- (b) Copy the specified entity with relationships, in which case the specified existing entity shall be copied to the specified new entity, along with all relationships in which the existing entity participates.
- (c) Display validation information, in which case the desired validation data and entity naming rules shall appear in the IRD Schema Area.
- (d) Refresh the Data Area, in which case the contents of the Data Area shall be returned to the form in which it originally appeared before user input.

#### 8.4.2 IRD Output Panel Trees

**Function** To identify all panel trees which produce output on the contents of the IRD, and the rules which apply to all such panel trees.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

IRD-output-panel-tree ::=
    general-output-panel-tree           8.4.2.1
    | output-impact-of-change-panel-tree 8.4.2.2
    | output-syntax-panel-tree           8.4.2.3
    | inquire-entity-panel-tree          8.4.2.4
    | inquire-relationship-panel-tree    8.4.2.5

```

**Syntax Rules**

- (1) The Syntax Rules which shall exist for user-supplied information in the Data Area shall include those which exist in the specifications for the equivalent functions in the Command Language Interface, except that the Syntax Rules which relate to command-clause ordering within a particular command specification shall not apply. Refer to Subsection 5.2.2.

**General Rules**

- (1) The General Rules which shall exist for these panel trees shall include those which exist for the equivalent command, with all of its associated command-clauses, in the Command Language Interface. Refer to Subsection 5.2.2.

**Security Rules** None.

**Actions**

- (1) The Actions which shall be performed by this panel tree shall be equivalent to those which are performed by the equivalent command in the Command Language Interface. Refer to Subsection 5.2.2 for a description of these actions.

**Error and Warning Conditions**

- (1) Error and Warning Conditions which can appear in the Message Area of one or more of the panels in the panel trees described in the following Subsections are the same as those which exist for the equivalent IRD output command found in Subsection 5.2.2.

#### 8.4.2.1 General Output Panel Tree

**Function** This panel tree shall be used to retrieve IRD descriptors, format output, and route the output to a specified destination.

##### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The IRD-View(s) to which the output shall be restricted. The entry of this information shall be optional, and if not entered, the IRD-View used shall be the default IRD-View in effect.
  - (b) The selection criteria for the entities to be included in the output.
  - (c) The sort parameters for sorting the set of entities into the order they are to appear in the output. The entry of this information shall be optional.
  - (d) The show options, which determine which IRD information shall displayed in the output for the selected entities.
  - (e) The destination to which the output shall be routed. The entry of this information shall be optional. If not entered, an implementation-specified default destination shall be used.

**IRD Schema Area Contents** None.

##### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Display the specified output, in which case the specified output shall be routed to the selected destination.
  - (b) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

#### 8.4.2.2 Output Impact-of-Change Panel Tree

**Function** This panel tree shall be used to identify and retrieve those entities which are potentially impacted by changes to selected entities.

##### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The impact option for determining those potentially-impacted entities to be displayed, either CUMULATIVE or INDIVIDUAL.

- (b) The IRD-View(s) to which the output shall be restricted. The entry of this information shall be optional, and if not entered, the IRD-View used shall be the default IRD-View in effect.
- (c) The selection criteria for the entities whose impact-of-change shall be analyzed.
- (d) The sort parameters for sorting the set of entities into the order in which the impact-of-change analysis shall be output. The entry of this information shall be optional.
- (e) The impact show options, which determine which IRD information shall be displayed in the output for the selected entities.
- (f) The destination to which the output shall be routed. The entry of this information shall be optional. If not entered, an implementation-specified default destination shall be used.

**IRD Schema Area Contents**     None.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Display the specified output, in which case the output shall be routed to the selected destination.
  - (b) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

#### **8.4.2.3 Output Syntax Panel Tree**

**Function**     This panel tree shall be used to display selected contents of the IRD in IRD command language syntax. Existence of this panel tree shall only be required in the event that the Command Language Interface is available in conjunction with the Panel Interface.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The IRD-View(s) to which the output shall be restricted. The entry of this information shall be optional, and if not entered, the IRD-View used shall be the default IRD-View in effect.
  - (b) The selection criteria for the entities to be output by this panel tree.
  - (c) The sort parameters for sorting the set of entities into the order in which the output shall appear. The entry of this information shall be optional.
  - (d) The syntax show options, which determine which IRD information shall be displayed in the output for the selected entities.

- (e) The destination to which the output shall be routed. The entry of this information shall be optional. If not entered, a default destination shall be used.

**IRD Schema Area Contents** None.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Display the specified output, in which case the output shall be routed to the selected destination.
  - (b) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

**8.4.2.4 Inquire Entity Panel Tree**

**Function** This panel tree shall be used to display an existing entity as the modify entity panel tree displays it, disallowing updates.

**Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the entity-access-name of an existing entity. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number of the entity.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the IRD-partition in which the specified entity exists, and all attributes and attribute-groups which are or can be assigned to the entity being displayed.

**IRD Schema Area Contents**

- (1) The IRD Schema Area of the panel(s) in this panel tree shall provide the rules for entity naming.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Display the specified entity, in which case the specified entity shall be displayed.
  - (b) Display rules for entity naming, in which case the desired information shall appear in the IRD Schema Area.

**8.4.2.5 Inquire Relationship Panel Tree**

**Function** This panel tree shall be used to display a relationship as the modify relationship panel tree displays it, disallowing any update.

**Data Area Contents**

- (1) The Data Area of the panel(s) in this panel tree shall request input of the following information:
  - (a) The entity-access-names of two existing entities in the IRD. If a full version-identifier is not specified by the user, then the system shall provide and display the default variation-name and revision-number assigned to the entities.
  - (b) The name of a relationship-type or relationship-class-type.
- (2) The Data Area of the panel(s) in this panel tree shall display, but shall not allow updates to, the IRD-partition in which each of the specified entities exist, and all attributes and attribute-groups which are or can be associated with the relationship being displayed.

**IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid relationship-type names and/or valid relationship-class-type names.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of the panel(s) in this panel tree shall provide to the user the following options:
  - (a) Display the specified relationship, in which case the attributes and attribute-groups of the specified meta-relationship are displayed.
  - (b) Display validation information and rules, in which case the desired validation criteria and rules shall appear in the IRD Schema Area.

**8.5 IRD-IRD Interface Panel Trees**

**Function** To identify all panel trees which support extracting the IRD Schema and contents of one IRD and incorporating them into another IRD.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

ird-ird-interface-panel-tree ::=
    create-IRD-panel-tree           8.5.1
    | export-IRD-panel-tree         8.5.2
    | check-IRD-schema-compatibility-panel-tree 8.5.3
    | import-IRD-panel-tree         8.5.4
    
```

### **Syntax Rules**

- (1) The Syntax Rules which shall exist for user-supplied information in the Data Area shall include those which exist in the specifications for the equivalent functions in the Command Language Interface, except that the Syntax Rules which relate to command-clause ordering within a particular command specification shall not apply. Refer to the IRD-IRD Interface Commands, Subsection 5.3.1, for a description of these rules.

### **General Rules**

- (1) The General Rules which shall exist for these panel trees shall include those which exist for the equivalent command, with all of its associated command-clauses, in the Command Language Interface. Refer to Subsection 5.3.1 for a description of these rules.

**Security Rules**     None.

### **Actions**

- (1) The Actions which shall be performed by this panel tree shall be equivalent to those which are performed by the equivalent command in the Command Language Interface. Refer to Subsection 5.3.1.

### **Error and Warning Conditions**

- (1) Error and Warning Conditions which can appear in the Message Area of one or more of the panels in the panel trees described in the following Subsections are the same as those which exist for the equivalent IRD maintenance command found in Subsection 5.3.1.

#### **8.5.1 Create IRD Panel Tree**

**Function**     This panel tree shall be used to create a new IRD of a specified name with an initial IRD Schema (and optionally load it with the contents of an IRD-export-file **\*\*EX/IM\*\***).

### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the new IRD to be created. This name shall not be the name of an existing IRD.
  - (b) The location at which to place the new IRD. Entry of this information shall be optional.
  - (c) The source of the IRD Schema to be used in the new IRD. This can optionally be the name of an IRD-schema-export-file, the name of an existing IRD, or an indication that the Minimal IRD Schema shall be used.
  - (d) The name and location of an IRD-export-file whose contents are to be loaded into the new IRD. The entry of this information shall be optional.

**IRD Schema Area Contents**     None.



**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display an option to create the new IRD, in which case a new IRD with the specified IRD Schema shall be created. If requested, this new IRD shall be loaded with the contents of the specified IRD-export-file.

**8.5.2 Export IRD Panel Tree \*\*EX/IM\*\***

**Function** This panel tree shall be used to obtain the IRD Schema and a subset of the IRD contents from an IRD and place them in specified data files in IRD Schema export format and IRD export format, respectively.

**Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The IRD-View(s) to which the IRD extraction shall be restricted. The entry of this information shall be optional, and if not entered, the IRD-View used shall be the default IRD-View in effect.
  - (b) The specification of the entities which are to be extracted from the current IRD, or an indication that only the IRD Schema shall be extracted. In the latter case, no IRD-export-file shall be produced.
  - (c) The types of relationships which are to be excluded from those relationships being extracted. Entry of this information shall be optional, and in the case where no relationship-type restrictions are specified, all relationships for which both members exist in the specified set of entities are extracted.
  - (d) The name and location of the IRD-schema-export-file into which the IRD Schema of the current IRD shall be placed. Entry of the location shall be optional.
  - (e) The name and location of the IRD-export-file into which the specified IRD entities and relationships are to be placed. Entry of this information shall be optional. In the case where the user has indicated that only the IRD Schema shall be extracted, this information shall not be entered.
  - (f) A File Title suffix to be appended as a suffix on the File Title in both the IRD-schema-export-file and the IRD-export-file. Entry of this information shall be optional.

**IRD Schema Area Contents** None.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Export the IRD, in which case two files, the IRD-schema-export-file and the IRD-export-file, are produced, unless the user has indicated that only the IRD Schema shall be created. In that event, only the IRD-schema-export-file shall be produced. The IRD-schema-export-file shall be loaded with the total IRD Schema of the IRD in

which the user is operating, regardless of IRD Schema life-cycle-phase. The IRD-export-file shall be loaded with the specified IRD entities and relationships.

- (b) Export Controlled Only, in which case the same actions as those provided above shall occur, except that only those IRD Schema descriptors in the CONTROLLED IRD Schema life-cycle-phase shall be extracted.

### 8.5.3 Check IRD Schema Compatibility Panel Tree \*\*EX/IM\*\*

**Function** This panel tree shall be used to determine if the IRD Schema which either

- (1) Exists for another IRD, or
- (2) Is contained in an IRD-schema-export-file,

is compatible with the IRD Schema of the IRD in which the user is operating. If the IRD Schema is compatible with the IRD Schema of the IRD in use, IRD descriptors can be transferred from one IRD to another using other IRD-IRD Interface panel trees.

#### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The source of the other IRD Schema to be used in the compatibility check. This can optionally be the name of an IRD-schema-export-file, the name of an existing IRD, or an indication that the Minimal IRD Schema shall be used.
  - (b) The designation of whether the other IRD Schema specified above shall be treated as the source IRD Schema or the target IRD Schema in the compatibility check. Designating the other IRD Schema as the source IRD Schema implies that the IRD Schema of the IRD in use shall be regarded as the target IRD Schema. Designating the other IRD Schema as the target IRD Schema implies that the IRD Schema of the IRD in use shall be regarded as the source IRD Schema.

**IRD Schema Area Contents** None.

#### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display an option to check compatibility, in which case the source IRD Schema shall be checked for compatibility with the target IRD Schema. If the two IRD Schemas are compatible, a notification of this shall be issued to the user via the data area of a panel in this panel tree. If any incompatibilities are detected, appropriate notification of these incompatibilities shall also be provided to the user via the data area of a panel in this panel tree.

### 8.5.4 Import IRD Panel Tree \*\*EX/IM\*\*

**Function** This panel tree shall be used to import the contents of an IRD-export-file into an IRD.

**Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name and location of the IRD-schema-export-file associated with the IRD-export-file which shall be imported.
  - (b) The name and location of the IRD-export-file whose contents are to be imported into the IRD currently in use.
  - (c) The designation of an empty, UNCONTROLLED IRD life-cycle-phase into which the contents of the IRD-export-file shall be imported.

**IRD Schema Area Contents** None.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4, the Action Area of one or more panels in this panel tree shall display an option to import the IRD, in which case the specified IRD-schema-export-file shall be checked for compatibility against the IRD in which the user is operating. If the two IRD Schemas are compatible, the contents of the specified IRD-export-file shall be imported into the empty, UNCONTROLLED IRD life-cycle-phase specified.

**9 The IRD Schema**

This section presents the specifications for the IRD Schema for Module 1 of the standard IRDS.

**9.1 Meta-Entity Types**

The following meta-entity-types exist in the structure of the IRD Schema:

- (1) **ENTITY-TYPE.** A meta-entity of this type corresponds to an entity-type, instances of which are in the IRD.
- (2) **RELATIONSHIP-TYPE.** A meta-entity of this type corresponds to a relationship-type, instances of which are in the IRD.
- (3) **RELATIONSHIP-CLASS-TYPE.** A meta-entity of this type defines a simplified form for specifying relationships and inverse relationships of one or more types.
- (4) **ATTRIBUTE-TYPE.** A meta-entity of this type corresponds to an attribute-type, instances of which are in the IRD.
- (5) **ATTRIBUTE-GROUP-TYPE.** A meta-entity of this type corresponds to an attribute-group-type (an ordered tuple of attribute-types), instances of which are in the IRD.

- (6) **ATTRIBUTE-TYPE-VALIDATION-DATA.** A meta-entity of this type provides specifications of valid value sets for one or more given attribute-types. Each meta-entity of this type is in the form suitable for a specific attribute-type-validation-procedure.
- (7) **ATTRIBUTE-TYPE-VALIDATION-PROCEDURE.** A meta-entity of this type represents a validation procedure that exists for the validation of attributes of one or more given attribute-types. When the IRD Schema is created, exactly two meta-entities of this type shall exist: **VALUE-VALIDATION** and **RANGE-VALIDATION**.
- (8) **IRD-PARTITION.** A meta-entity of this type defines a logical partition for a set of IRD entities. Every IRD entity shall be in one and only one IRD-PARTITION. There shall exist one IRD-PARTITION meta-entity named **SECURITY**. Other IRD-PARTITION meta-entities shall define IRD life-cycle-phases. Each of these meta-entities shall have a **LIFE-CYCLE-PHASE-CLASS** meta-attribute. These meta-attributes are system-maintained.

When an IRD Schema is created, three life-cycle phases and the IRD-Partition named **SECURITY** shall be created. The life-cycle phases are **UNCONTROLLED-LIFE-CYCLE-PHASE**, **CONTROLLED-LIFE-CYCLE-PHASE**, and **ARCHIVED-LIFE-CYCLE-PHASE**. These have the **LIFE-CYCLE-PHASE-CLASS** meta-attributes equal to **UNCONTROLLED**, **CONTROLLED**, and **ARCHIVED**, respectively. Any IRD partition created by the IRD Schema maintenance facility shall be given meta-attribute **LIFE-CYCLE-PHASE-CLASS** equal to **UNCONTROLLED**.

The IRD-Partitions **SECURITY**, **CONTROLLED-LIFE-CYCLE-PHASE** and **ARCHIVED-LIFE-CYCLE-PHASE** each shall have the **SYSTEM-LOCK** meta-attribute equal to **ON**. Thus, they shall not be deleted.

- (9) **QUALITY-INDICATOR.** A meta-entity of this type defines a means for identifying entities according to some condition or qualitative assessment.
- (10) **VARIATION-NAMES-DATA.** A meta-entity of this type defines a set of variation-names which can be used in naming entities of one or more entity-types.
- (11) **IRDS-LIMITS.** There shall be one meta-entity of this type in the IRD Schema, named **EXISTING-IRDS-LIMITS**. It documents implementation-defined limits. The meta-attributes of this meta-entity are established by the implementation. The meta-entity shall be system created and shall not be deleted or modified by any user command.
- (12) **IRDS-DEFAULTS.** There shall be one meta-entity of this type, named **EXISTING-IRDS-DEFAULTS**. It shall be system-created. This meta-entity specifies defaults for meta-attributes. These defaults are initially established by an implementation, but can be modified by an installation. The meta-entity shall not be deleted.
- (13) **IRDS-RESERVED-NAMES.** The meta-entity **STANDARD-RESERVED-NAMES** defines the assigned-access-names of those entities and meta-entities that are required by the IRDS. An implementation can create other meta-entities of this type, but they shall not be named **STANDARD-RESERVED-NAMES**.
- (14) **NAMES.** There shall be only one meta-entity of this type. It shall be given the name **NAMING-RULES**. This meta-entity contains a description of the rules for naming both entities and meta-entities. It is intended for user documentation. It shall be system-created and not modifiable. It contains one text meta-attribute, which

describes the naming rules defined in Subsection 4.4. The description of the rules defined in Subsection 4.4 is left to the implementation.

## 9.2 Meta-Relationship Types

A meta-relationship is a directed association between two meta-entities. A meta-relationship-type is a set of meta-relationships for which:

- (1) The type of the first and second meta-entities is defined.
- (2) There is a label (called the meta-relationship-class-type) which expresses the nature of the association.
- (3) Sequencing and uniqueness rules are defined.

The following meta-relationship-class-types are defined:

MEMBER-OF  
CONNECTS  
CONTAINS  
USES

The name of a meta-relationship-type consists of the first meta-entity-type, the meta-relationship-class-type, the second meta-entity-type, all connected by a "-".

The following meta-relationship-types are defined for the IRD Schema:

RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-CLASS-TYPE  
RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE  
RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE  
RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE  
ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE  
ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE  
ENTITY-TYPE-USES-VARIATION-NAMES-DATA  
ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE  
ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA  
ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-PROCEDURE

Consult the Table in Subsection 9.5 for the definition of the correspondence between meta-relationship-types and meta-attribute-types.

The following Subsections 9.2.1 through 9.2.4 define integrity constraints for meta-relationships. Subsection 9.2.5 defines each meta-relationship-type.

### 9.2.1 Unique and Nonunique Meta-Relationship Types

It shall be possible to have the same ordered pair of meta-entities in more than one meta-relationship of the meta-relationship-type RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE. This meta-relationship-type is said to be nonunique. A key meta-attribute shall be required to identify a specific meta-relationship of that type. This key meta-attribute-type is POSITION.

For all other meta-relationship-types, there shall exist only one meta-relationship with a given ordered pair of meta-entities. These meta-relationship-types do not have a key meta-attribute-type, and are said to be unique.

### 9.2.2 Sequenced and Nonsequenced Meta-Relationship Types

The following meta-relationship-types are called sequenced:

RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE  
 ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE

For sequenced meta-relationship-types, a meta-attribute-type is designated as the sequencing meta-attribute-type. For any given meta-entity-1, all meta-relationships have a unique sequencing meta-attribute.

POSITION is the sequencing meta-attribute-type for the meta-relationship-type RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE. Note that POSITION is also a key meta-attribute-type in this meta-relationship-type. GROUP-POSITION is the sequencing meta-attribute-type for the meta-relationship-type ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE. Note here that the sequencing meta-attribute is not a key meta-attribute-type.

Thus, every nonunique meta-relationship-type is sequenced, and the sequencing meta-attribute-type is the key meta-attribute-type for the meta-relationship-type. However, there can be sequenced meta-relationship-types which are also unique, as in the case of ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE.

For sequenced meta-relationship-types which are unique with respect to meta-entities, the sequencing meta-attribute is used to support IRD Schema output. It need not be specified to identify a given meta-relationship.

All other meta-relationship-types are non-sequenced.

### 9.2.3 Meta-Relationship Type Ratios

Each meta-relationship-type used in the IRD Schema has a ratio, defined as follows:

Let  $A, B$  be meta-entity-types.  
 Let  $a, a'$  be meta-entities of type  $A$ .  
 Let  $b, b'$  be meta-entities of type  $B$ .  
 Let  $(A,X,B)$  denote the meta-relationship-type with:  
      $A$  as meta-entity-type-1, and  
      $B$  as meta-entity-type-2.  
 Let  $X$  be an association-label.  
 Let  $(a,X,b)$  denote a meta-relationship of type  $(A,X,B)$ .

- (1) If  $(A,X,B)$  has ratio  $(0,n:0,1)$ , then:
  - (a) For any given  $a$  there shall be 0 or 1 meta-relationships of the form  $(a,X,b')$ , and
  - (b) For any given  $b$ , there shall be 0, 1, or more meta-relationships of the form  $(a'X,b)$ .
- (2) If  $(A,X,B)$  has ratio  $(0,n:2)$ , then:
  - (a) For any given  $a$  there 2 meta-relationships of the form  $(a,X,b')$ , and
  - (b) For any given  $b$  there shall be 0, 1, or more meta-relationships of the form  $(a'X,b)$ .
- (3) If  $(A,X,B)$  has ratio  $(0,n:0,m)$ , then:
  - (a) For any given  $a$  there shall be 0, 1, or more meta-relationships of the form  $(a,X,b')$ , and
  - (b) For any given  $b$  there shall be 0, 1, or more meta-relationships of the form  $(a'X,b)$ .

The following meta-relationship-types have ratio (0,n:0,1):

RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-CLASS-TYPE  
 ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-PROCEDURE

The following meta-relationship-type has ratio (0,n:2):

RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE

The following meta-relationship-types have ratio (0,n:0,m):

RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE  
 RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE  
 ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE  
 ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE ENTITY-TYPE-USES-VARIATION-  
 NAMES-DATA  
 ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE  
 ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA

#### 9.2.4 The IRD Schema Life-Cycle-Phase Integrity Rule

All meta-relationships conform to the following rule, called the IRD Schema life-cycle-phase integrity rule:

Let  $a$  and  $b$  be meta-entities of types  $A$  and  $B$ , respectively.  
 Let  $(a,X,b)$  denote a meta-relationship of type  $(A,X,B)$ .

Then:

- (1) If  $a$  is in the UNCONTROLLED IRD Schema life-cycle-phase,  $b$  shall also be in either the UNCONTROLLED or CONTROLLED IRD Schema life-cycle-phase.
- (2) If  $a$  is in the CONTROLLED IRD Schema life-cycle-phase,  $b$  shall also be in the CONTROLLED IRD Schema life-cycle-phase.
- (3) If  $a$  is in the ARCHIVED IRD Schema life-cycle-phase,  $b$  shall be in either the ARCHIVED or CONTROLLED IRD Schema life-cycle-phase.

#### 9.2.5 Meta-Relationship Type Definitions

A discussion of each of the meta-relationship-types follows. Any constraints in addition to those described in 9.2.1 through 9.2.4 are identified in the discussion for each meta-relationship-type. Where appropriate, a discussion of how meta-relationships of the specified type support the functionality of the IRD command language is provided within comments. (Comments are delimited by "/\*" and "\*/". These are for clarification only and are not part of this standard.)

- (1) RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-CLASS-TYPE. This meta-relationship-type associates a relationship-type meta-entity to a relationship class-type meta-entity. Any relationship-type meta-entity shall be associated with no more than one relationship-class-type meta-entity.

/\* This meta-relationship-type is intended to facilitate simplified coding of relationships. Recall that the identification of a relationship has two forms. The first form is:

entity-1-access-name  
relationship-type-designator  
entity-2-access-name

and the second form is:

entity-1-access-name  
relationship-class-type-designator  
entity-2-access-name.

As an example, consider the relationship-type IRDS-USER-HAS-IRD-VIEW within the Minimal IRD Schema. Suppose *X* is an IRDS-USER entity and *Y* is an IRD-VIEW entity in the IRD. The first form, X IRDS-USER-HAS-IRD-VIEW Y, requires more keystrokes than X HAS Y.

To the IRDS user entering the command, the second form is clearer. The existence of a meta-relationship from the relationship-type IRDS-USER-HAS-IRD-VIEW to the relationship-class-type HAS enables the IRDS command language processor to determine that the shorter form does specify a valid relationship. Without this meta-relationship, the short form would not be regarded as valid by the IRDS.

Also, within the selection criteria, specifying the name of relationship-class-type within a path-list is equivalent to specifying associated relationship-types as paths. Thus the syntax:

```
SELECT ENTITIES DIRECTLY RELATED TO X VIA HAS
```

is equivalent to specifying the syntax:

```
SELECT ENTITIES DIRECTLY RELATED TO X VIA  
IRDS-USER-HAS-IRD-VIEW,  
IRDS-USER-HAS-IRD-SCHEMA-VIEW
```

\*/

- (2) **RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE.** This meta-relationship-type shall be used to define which entity-types are associated by a relationship-type. For any given relationship-type, two meta-relationships of this type are required for successful installation of the relationship-type.

The POSITION meta-attribute-type shall be associated with this meta-relationship-type shall be used to sequence these meta-relationships.

Sequencing meta-attributes are part of the identification of meta-relationships. Thus, there can be multiple meta-relationships of this type between the same two meta-entities.

- (3) **RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE.** A meta-relationship of this type associates a relationship-type with an attribute-group-type. This meta-relationship-type has ratio (0,n:0,m). There shall be no associated sequencing meta-attribute-type. Thus, only one meta-relationship of this type shall exist for any given pair of relationship-type and attribute-group-type meta-entities.



- (4) **RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE.** A meta-relationship of this type associates a relationship-type with an attribute-type. This meta-relationship-type has ratio (0,n:0,m). There shall be no associated sequencing meta-attribute-type. Thus, only one meta-relationship of this type shall exist for any given pair of relationship-type and attribute-type meta-entities.
- (5) **ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE.** A meta-relationship of this type associates an entity-type with an attribute-group-type. This meta-relationship-type has ratio (0,n:0,m). There shall be no associated sequencing meta-attribute-type. Thus, only one meta-relationship of this type shall exist for any given pair of entity-type and attribute-group-type meta-entities.
- (6) **ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE.** A meta-relationship of this type associates an entity-type with an attribute-type. This meta-relationship-type has ratio (0,n:0,m). There shall be no associated sequencing meta-attribute-type. Thus, only one meta-relationship of this type shall exist for any given pair of entity-type and attribute-type meta-entities.
- (7) **ENTITY-TYPE-USES-VARIATION-NAMES-DATA.** A meta-relationship of this type associates an entity-type with a variation-names-data meta-entity. This meta-relationship-type has ratio (0,n:0,m). Only one meta-relationship of this type shall exist for any given pair of entity-type and variation-names-data meta-entities.
- /\* Note that the set of valid variation-names for any given entity-type consists of the union of all variation-names within all variation-names-data meta-entities associated with that entity-type. \*/
- (8) **ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE.** A meta-relationship of this type defines the attribute-types which constitute a given attribute-group-type. This meta-relationship-type has ratio (0,n:0,m). Only one meta-relationship of this type shall exist for any given pair of attribute-group-type and attribute-type meta-entities.
- A conformant implementation of the standard IRDS may disallow the case where an attribute-type meta-entity with a **FORMAT** meta-attribute equal to **TEXT** participates in a meta-relationship of this type.
- The meta-attribute-type **GROUP-POSITION** shall be associated with this meta-relationship-type. Note that it is not a sequencing meta-attribute-type for this meta-relationship-type. It is subject to the additional constraint that for any particular attribute-group-type meta-entity, all **GROUP-POSITION** meta-attributes shall be distinct.
- (9) **ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-PROCEDURE.** A meta-relationship of this type identifies the kind of validation which shall be performed on any given attribute-type. This meta-relationship-type has ratio (0,n:0,1). For any given attribute-type meta-entity, this meta-relationship shall not be deleted unless there is no meta-relationship from the attribute-type to any attribute-type-validation-data meta-entity.
- (10) **ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA.** A meta-relationship of this type identifies the domain of a given attribute-type. This meta-relationship-type has ratio (0,n:0,m).

For any given attribute-type meta-entity, meta-relationships of this type can be established only if there exists a meta-relationship from the attribute-type to an attribute-type-validation-procedure meta-entity.

If the specified attribute-type is associated with the attribute-type-validation-procedure VALUE-VALIDATION, then the specified attribute-type-validation-data meta-entity shall have its meta-attribute VALIDATION-TYPE equal to VALUE.

If the specified attribute-type is associated with the attribute-type-validation-procedure RANGE-VALIDATION, then the specified attribute-type-validation-data meta-entity shall have its meta-attribute VALIDATION-TYPE equal to RANGE.

### 9.3 Meta-Attribute Types

To define each meta-attribute-type completely, the following information is presented for each meta-attribute-type:

- (1) **Name.** The name is the IRD Schema language token used to identify the meta-attribute as identified in Subsection 4.7. The name is the heading used to identify the meta-attribute-type.
- (2) **Description.** A textual description of the meta-attribute-type.
- (3) **Valid Values.** Valid values can be specified either as a list of discrete values or as one or more ranges of values. In such cases, the meta-attribute-type is called constrained. If nothing is specified here, the meta-attribute-type is unconstrained. All text meta-attribute-types are unconstrained.
- (4) **Default.** If the meta-attribute-type is required for any meta-entity-type or meta-relationship-type, a default value is given. Otherwise, none is given.
- (5) **Uniqueness Rules.** If a "Yes" is given here, then the rules are given in the Constraints Section. If a "No" is given, there are no name uniqueness restrictions.
- (6) **Constraints.** Any dependencies between the value of a meta-attribute of the specified type and other IRD Schema descriptors are identified here.
- (7) **Format.** Each meta-attribute-type shall have one of three possible formats: numeric (N), string (S), or text (T). Numeric meta-attribute-types have valid values which can be specified as unsigned integers. String meta-attribute-types have values which can be either irds-names or short-string-literals. Text meta-attribute-types are composed of one to an implementation-dependent maximum number of lines of text. Each line of text has a unique line-number.
- (8) **Minimum Length.** The minimum number of characters for any valid value for the meta-attribute. This is specified only if the format is numeric or string.
- (9) **Maximum Length.** The maximum number of characters for any valid value for the meta-attribute. This is specified only if the format is numeric or string. If "IDEP" is specified below, the maximum length is Implementation-DEPENDent.

- (10) **Repeating.** This property of the meta-attribute-type identifies whether multiple values can be assigned (indicated by a "Y"), or only a single value can be assigned (indicated by an "N").
- (11) **System Maintained.** This property of the meta-attribute-type identifies whether the assigned value(s) is maintained by the IRDS ("Y") or by the installation ("N").
- (12) **Fixed.** A "Y" here indicates that once established by the implementor, the meta-attribute is fixed in the IRDS, meaning that it shall not be changed by an installation. An "N" means that the value is not fixed.
- (13) **Required.** A "Y" for this property indicates that if the meta-attribute-type is associated with a meta-entity-type or meta-relationship-type, then a meta-attribute shall exist for each corresponding meta-entity or meta-relationship. This means that the meta-attribute can be modified, but shall never be deleted. An "N" indicates that the meta-attribute-type is not required.

Meta-attribute-types are presented below in alphabetical order:

**Added By**

Name: ADDED-BY  
 Description: Identifies the IRDS-USER who was active when the corresponding meta-entity or meta-relationship was added to the IRD Schema.  
 Valid Values: Installation Specified.  
 Default: None.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	N	Y	N	Y

**Common To Entity-Types**

Name: COMMON-TO-ENTITY-TYPES  
 Description: Identifies whether or not an attribute-type or attribute-group-type shall be common to all entity-types.  
 Valid Values: YES, NO  
 Default: NO  
 Uniqueness Rules: No.  
 Constraints: If the meta-attribute equals YES, then the attribute-type or attribute-group-type meta-entity shall be associated with each entity-type meta-entity when the entity-type is added to the IRD Schema. Any attribute-type or attribute-group-type meta-entity which has this meta-attribute equal to YES shall not be specified within any add-meta-relationship or delete-meta-relationship command.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	Y	Y	N

**Connectable**

Name: CONNECTABLE  
 Description: Identifies whether or not the specified entity-type meta-entity can participate in any user-defined relationship-types.  
 Valid Values: YES, NO  
 Default: YES  
 Uniqueness Rules: No.  
 Constraints: None.  
 /\* It is possible for an entity-type which is not connectable to participate in relationship-types which are either part of the Minimal IRD Schema or provided by the implementor. \*/

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	N	Y	N

**Decoded Value**

Name: DECODED-VALUE  
 Description: A value which corresponds to an encoded value for an attribute-type, but which is more descriptive than the encoded value. The decoded value is intended to be used as the display value for output.  
 Valid Values: Installation specifiable.  
 Default: None.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	Y	N	N	N

**Description Of Rules**

Name: DESCRIPTION-OF-RULES  
 Description: A description of the Naming-Rules in existence for entity and meta-entity names in the standard IRDS.  
 Valid Values: Installation specifiable.  
 Default: None.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
T	N/A	N/A	N	N	N	N

**Encoded Value**

Name: ENCODED-VALUE  
 Description: An encoded value is a valid input value for an attribute-type; i.e., it can be specified for an attribute of that type within IRD "commands".  
 Valid Values: Installation specifiable.  
 Default: None.  
 Uniqueness Rules: No.

Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	Y	N	N	N

**Entity Class**

Name: ENTITY-CLASS  
 Description: Identifies the basic usage of the corresponding entity-type.  
 Valid Values: DATA, EXTERNAL, PROCESS, SECURITY  
 Default: None.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	4	8	N	N	N	N

**Format**

Name: FORMAT  
 Description: The basic format for an attribute-type within IRD "commands."  
 Valid Values: STRING, TEXT, INTEGER, REAL, DATE, TIME  
 /\* STRING identifies either a quoted string or an irds-name format. TEXT identifies an attribute-type whose values consist of one or more lines of text. INTEGER identifies a signed or unsigned integer. REAL identifies an approximate numeric value. DATE identifies the mask CCYYMMDD. TIME identifies the mask HHMMSS. \*/  
 Default: STRING  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	4	7	N	N	N	Y

**Group Position**

Name: GROUP-POSITION  
 Description: Identifies the relative position of an attribute-type within an associated attribute-group-type.  
 Valid Values: Implementation-dependent.  
 Default: None.  
 Uniqueness Rules: No.  
 Constraints: For any given ATTRIBUTE-GROUP-TYPE meta-entity, all GROUP-POSITION meta-attributes shall be distinct within the set of all meta-relationships of type ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEP	N	N	N	N

### High Value

**Name:** HIGH-VALUE  
**Description:** The high data value of a DATA-RANGE interval.  
**Valid Values:** Installation specifiable.  
**Default:** None.  
**Uniqueness Rules:** No.  
**Constraints:** This value shall be consistent with the FORMAT meta-attribute of the associated attribute-type meta-entities, and shall be greater than or equal to the corresponding LOW-VALUE meta-attribute of the DATA-RANGE interval.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	N	N	N	N

### Implementation Lock

**Name:** IMPLEMENTATION-LOCK  
**Description:** Identifies those meta-entities which are required for an implementation.  
**Valid Values:** ON, OFF  
**Default:** OFF  
**Uniqueness Rules:** No.  
**Constraints:** None

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	Y	N	Y

### Integer Limit

**Name:** INTEGER-LIMIT  
**Description:** Defines the maximum integer value allowed for numeric attribute-types.  
**Valid Values:** An integer not less than 32767.  
**Default:** 32767  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	5	IDEP	N	Y	Y	Y

### Inverse Name

**Name:** INVERSE-NAME  
**Description:** The name given to the inverse of a relationship-type. Since a relationship-type is a directed association between two entity-types, the inverse of a relationship-type is the association corresponding to the "opposite" direction.  
**Valid Values:** Any valid irds-name.  
**Default:** None.  
**Uniqueness Rules:** Yes.  
**Constraints:** An inverse-name shall only be associated with either a relationship-type or relationship-class-type meta-entity.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	N	N	N	N

**Last Modified By**

**Name:** LAST-MODIFIED-BY  
**Description:** Identifies the IRDS-USER who was active when the corresponding meta-entity or meta-relationship was last modified.  
**Valid Values:** Installation specifiable.  
**Default:** None.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	N	Y	N	N

**Life Cycle Phase Class**

**Name:** LIFE-CYCLE-PHASE-CLASS  
**Description:** Defines whether a particular LIFE-CYCLE-PHASE is UNCONTROLLED, CONTROLLED, or ARCHIVED.  
**Valid Values:** UNCONTROLLED, CONTROLLED, ARCHIVED  
**Default:** UNCONTROLLED.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	8	12	N	Y	Y	N

**Line Count Limit**

**Name:** LINE-COUNT-LIMIT  
**Description:** Defines the maximum number of lines allowed for TEXT attributes.  
**Valid Values:** An integer not less than 32767.  
**Default:** 32767  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	5	IDEP	N	Y	Y	Y

**Line Length Limit**

**Name:** LINE-LENGTH-LIMIT  
**Description:** Defines the maximum width of a line of text.  
**Valid Values:** An integer not less than 72.  
**Default:** 72  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	Y	Y	Y

**Low Value**

Name: LOW-VALUE  
 Description: The low data value of a DATA-RANGE interval.  
 Valid Values: Unconstrained.  
 Default: None.  
 Uniqueness Rules: No.  
 Constraints: This value shall be consistent with the FORMAT meta-attribute of the associated attribute-type meta-entities, and shall be less than the corresponding HIGH-VALUE meta-attribute of the DATA-RANGE interval.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	N	N	N	N

**Maximum Attribute Length**

Name: MAXIMUM-ATTRIBUTE-LENGTH  
 Description: The maximum number of characters that can be specified in a string, or a line of text, or the maximum number of digits that can be specified in an integer or real, for any value of a given attribute-type.  
 Valid Values: 1 through the implementation-dependent maximum number of characters for short-string-literal (see Subsection 4.2).  
 Default: Implementation-dependent.  
 Uniqueness Rules: No.  
 Constraints: A meta-attribute of this type shall be valid only if the attribute-type FORMAT is not equal to DATE or TIME.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	3	N	N	N	N

**Maximum Entity Assigned Access Name Length**

Name: MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH  
 Description: The maximum number of characters which can be specified in the assigned-access-name for entities of a given entity-type.  
 Valid Values: An integer with a value of 1 through the implementation-dependent maximum number of characters for short-string-literal.  
 Default: The value of MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	3	N	N	N	N



**Maximum Entity Assigned Access Name Length Default**

Name: MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT  
 Description: The default value for MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH.  
 Valid Values: An integer with a value of 31 through the implementation-dependent maximum number of characters for short-string-literal.  
 Default: Implementation-dependent.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	N	N	Y

**Maximum Entity Assigned Access Name Length Limit**

Name: MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT  
 Description: Defines the maximum number of characters allowed by the implementation for an entity assigned-access-name.  
 Valid Values: An integer with value 31 through 255 inclusive.  
 Default: 255  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	Y	Y	Y

**Maximum Entity Assigned Descriptive Name Length**

Name: MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH  
 Description: The maximum number of characters which can be used in the assigned-descriptive-name for entities of a particular entity-type.  
 Valid Values: An integer with a value of 31 through an implementation-dependent maximum value.  
 Default: The value of MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	N	N	N

**Maximum Entity Assigned Descriptive Name Length Default**

Name: MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT  
 Description: The default value for MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH.  
 Valid Values: An integer with a value of 31 through an implementation-dependent maximum value.  
 Default: Implementation-dependent.  
 Uniqueness Rules: No.

Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	N	N	Y

**Maximum Entity Assigned Descriptive Name Length Limit**

Name: MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT  
 Description: Defines the maximum number of characters allowed by the implementation for an entity assigned-descriptive-name.  
 Valid Values: An integer with value 31 through 255 inclusive.  
 Default: 255  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	Y	Y	Y

**Maximum Meta-Entity Assigned Access Name Length Limit**

Name: MAXIMUM-META-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT  
 Description: Defines the maximum number of characters allowed by the implementation for a meta-entity assigned-access-name.  
 Valid Values: An integer with value 31 through 255 inclusive.  
 Default: 255  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	Y	Y	Y

**Maximum Meta-Entity Assigned Descriptive Name Length Limit**

Name: MAXIMUM-META-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT  
 Description: Defines the maximum number of characters allowed by the implementation for a meta-entity assigned-descriptive-name.  
 Valid Values: An integer with value 31 through 255 inclusive.  
 Default: 255  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	Y	Y	Y

**Maximum Number Of Occurrences**

Name: MAXIMUM-NUMBER-OF-OCCURRENCES  
 Description: The maximum number of occurrences allowed for a repeating attribute or attribute-group of a particular type.

**Valid Values:** An integer with a value of 2 through an implementation-dependent maximum value.  
**Default:** The value of MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEP	N	N	N	N

**Maximum Number Of Occurrences Default**

**Name:** MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT  
**Description:** The default for MAXIMUM-NUMBER-OF-OCCURRENCES.  
**Valid Values:** An integer with a value of 2 through an implementation-dependent maximum value.  
**Default:** Implementation-dependent.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEP	N	N	N	Y

**Maximum Number Of Occurrences Limit**

**Name:** MAXIMUM-NUMBER-OF-OCCURRENCES-LIMIT  
**Description:** The maximum number of occurrences allowed by an implementation for repeating attributes and attribute-groups.  
**Valid Values:** An integer not less than 256.  
**Default:** 256  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	3	IDEP	N	Y	Y	Y

**Meta-Entity Substitute Name**

**Name:** META-ENTITY-SUBSTITUTE-NAME  
**Description:** A string of characters which can be used in place of the meta-entity-access-name within a command syntax.  
**Valid Values:** Any valid irds-name.  
**Default:** None.  
**Uniqueness Rules:** Yes.  
**Constraints:** This name shall be unique within the IRD Schema. That is, it shall not be the same as the assigned-access-name, meta-entity-substitute-name, or inverse-name of any existing meta-entity.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	Y	N	N	N

**Minimum Attribute Length**

Name: MINIMUM-ATTRIBUTE-LENGTH  
 Description: The minimum number of characters which shall be specified for an attribute of a particular type.  
 Valid Values: An integer with a value of 1 through the value of MAXIMUM-ATTRIBUTE-LENGTH.  
 Default: Implementation-dependent.  
 Uniqueness Rules: No.  
 Constraints: Value shall not exceed the value of MAXIMUM-ATTRIBUTE-LENGTH.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEP	N	N	N	N

**Minimum Entity Assigned Access Name Length**

Name: MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH  
 Description: The minimum number of characters which shall be specified in the assigned-access-name for entities of a given entity-type.  
 Valid Values: An integer with a value of 1 through the implementation-dependent maximum number of characters for short-string-literal.  
 Default: The value of MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	3	N	N	N	N

**Minimum Entity Assigned Access Name Length Default**

Name: MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT  
 Description: The default value for MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH.  
 Valid Values: An integer with a value of 1 through the implementation-dependent maximum number of characters for short-string-literal.  
 Default: Implementation-dependent.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	3	N	N	N	Y

**Minimum Entity Assigned Descriptive Name Length**

Name: MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH  
 Description: The minimum number of characters which shall be used in the assigned-descriptive-name for entities of a particular entity-type.  
 Valid Values: An integer with a value of 1 through an implementation-dependent maximum value.

**Default:** The value of MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	3	N	N	N	N

**Minimum Entity Assigned Descriptive Name Length Default**

**Name:** MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT  
**Description:** The default value for MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH.  
**Valid Values:** An integer with a value of 1 through an implementation-dependent maximum value.  
**Default:** Implementation-dependent.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	3	N	N	N	Y

**Number Of Instances**

**Name:** NUMBER-OF-INSTANCES  
**Description:** For meta-entities of type ENTITY-TYPE, this meta-attribute identifies how many entities of the corresponding type exist in the IRD. For meta-entities of type RELATIONSHIP-TYPE, this meta-attribute identifies how many relationships of the corresponding type exist in the IRD. For meta-entities of ATTRIBUTE-GROUP-TYPE and ATTRIBUTE-TYPE, this meta-attribute identifies how many entities and/or relationships have attributes or attribute-groups of the corresponding type. For meta-entities of type IRD-PARTITION, this meta-attribute identifies how many entities exist in the IRD-PARTITION. For QUALITY-INDICATOR, this meta-attribute identifies how many entities are associated with the given QUALITY-INDICATOR.  
**Valid Values:** Any nonnegative integer up to an implementation-dependent maximum.  
**Default:** Zero.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEP	N	Y	N	Y

**Number Of Times Modified**

**Name:** NUMBER-OF-TIMES-MODIFIED  
**Description:** The number of times a particular meta-entity or meta-relationship has been modified.  
**Valid Values:** Any nonnegative integer up to an implementation defined maximum value.

Default: Zero.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEF	N	Y	N	N

**Origin**

Name: ORIGIN  
 Description: Identifies the source of the definition of this object type:  
 - an International Standard  
 - a National Standard  
 - A Vendor Extension.  
 - An Installation Extension.  
 Valid Values:  
 - For an International Standard: ISsuffix  
 - For an American National Standard: X3.suffix  
 - For any other National Standard: NSsuffix  
 - For a Vendor Extension: VXsuffix  
 - For an Installation Extension: IXsuffix  
 where in each case the suffix shall contain the required identification of the standard.  
 In this standard IRDS the valid values are of the form X3.138-1988 (Ch.x), where x is the number of the Module which is the source of the definition.  
 Default: IXsuffix, where suffix is an installation-defined suffix.  
 Any meta-entity created by an installation via an IRD Schema maintenance command shall have this meta-attribute equal to IXsuffix.  
 Uniqueness Rules: No.  
 Constraints: Descriptors specified by a vendor shall be assigned a value of VXsuffix.  
 A meta-attribute of this type shall be assigned to a meta-entity when the meta-entity is created.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	IDEF	IDEF	N	Y	Y	Y

**Picture**

Name: PICTURE  
 Description: Defines a valid sequence of characters for either an irds-name or string-literal. For entity-types, PICTURE attributes define valid formats for the entity-access-names. For attribute-types, they define valid formats for the values.  
 Valid Values: Each value is a contiguous sequence of the characters A, N, and X where:  
 - A denotes any letter.  
 - N denotes any digit.  
 - X denotes any letter, digit, spacing-character or naming-character.  
 Default: None.  
 Uniqueness Rules: No.

- Constraints:**
- (1) Any PICTURE which is longer than the maximum number of characters for the entity-access-names of the entity-type or the values allowed for the attribute-type is truncated to that maximum length.
  - (2) Any PICTURE which is shorter than the maximum number of characters is assumed to be padded with trailing X's.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	SSL(*)	Y(**)	N	N	N

(\* ==> SSL denotes the maximum value of Short-String-Literal. )  
 (\*\* ==> If the pictures repeat, the entity-access-name or value shall be regarded as valid if it conforms to any PICTURE occurrence for the corresponding entity-type or attribute-type. )

**Position**

**Name:** POSITION  
**Description:** The position of an entity-type with respect to a relationship-type.  
**Valid Values:** 1, 2  
**Default:** None.  
**Uniqueness Rules:** No.  
**Constraints:** This meta-attribute-type is associated with the meta-relationship-type which associates RELATIONSHIP-TYPE meta-entities with ENTITY-TYPE meta-entities. For all meta-relationships associated with a given RELATIONSHIP-TYPE meta-entity, the POSITION meta-attributes shall be distinct.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	1	N	N	N	N

**Purpose**

**Name:** PURPOSE  
**Description:** Provides a description of the intended use and functionality supported by the associated meta-entity.  
**Valid Values:** Installation specifiable.  
**Default:** None.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
T	N/A	N/A	N	N	N	N

**Reserved Entity Name**

**Name:** RESERVED-ENTITY-NAME  
**Description:** Identifies those assigned-access-names and assigned-descriptive-names for entities which are reserved by either this standard or the implementation.  
**Valid Values:** Implementation dependent.  
**Default:** None.

Uniqueness Rules: No.  
 Constraints: These are established by the system. Other Modules of the standard or the implementation may add to these values.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	*	Y	Y	Y	Y

(\* ==> The maximum length equals the maximum length of any entity's assigned-access-name.)

**Reserved Meta-Entity Name**

Name: RESERVED-META-ENTITY-NAME  
 Description: Identifies those assigned-access-names, assigned-descriptive-names, meta-entity-substitute-names, and inverse-names for meta-entities which are reserved by either this standard or the implementation.

Valid Values: EXISTING-IRDS-DEFAULTS,  
 EXISTING-IRDS-LIMITS,  
 NAMING-RULES,  
 STANDARD-RESERVED-NAMES,  
 RANGE-VALIDATION,  
 ARCHIVED-LIFE-CYCLE-PHASE,  
 CONTROLLED-LIFE-CYCLE-PHASE,  
 UNCONTROLLED-LIFE-CYCLE-PHASE,  
 SECURITY,  
 VALUE-VALIDATION,  
 IRDS-USER,  
 IRD-VIEW,  
 IRD-SCHEMA-VIEW,  
 IRDS-USER-HAS-IRD-VIEW,  
 IRDS-USER-HAS-IRD-SCHEMA-VIEW,  
 IRD-PARTITION-NAME,  
 IRD-SCHEMA-PHASE-NAME,  
 DEFAULT-VIEW,  
 ADDED-BY,  
 LAST-MODIFIED-BY,  
 NUMBER-OF-TIMES-MODIFIED,  
 SYSTEM-DATE,  
 SYSTEM-TIME,  
 YES-OR-NO-VALUE,  
 DATE-TIME-ADDED,  
 DATE-TIME-LAST-MODIFIED

Default: None.  
 Uniqueness Rules: No.  
 Constraints: These are established by the system. Other Modules of the standard or the implementation may add to the above list.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	*	Y	Y	Y	Y

(\* ==> The maximum length equals the maximum length of any meta-entity's assigned-access-name.)



**Sequence Parameter**

**Name:** SEQUENCE-PARAMETER  
**Description:** Identifies whether or not an attribute-type which is associated with a particular relationship-type is to be used to sequence relationships of that type.  
**Valid Values:** YES, NO  
**Default:** NO.  
**Uniqueness Rules:** No.  
**Constraints:** This meta-attribute type shall be specified if the meta-attribute-type SEQUENCED = YES.

If the meta-attribute-type SEQUENCED = YES for a given RELATIONSHIP-TYPE and SEQUENCE-PARAMETER equals NO for every meta-relationship between the given RELATIONSHIP-TYPE meta-entity and an ATTRIBUTE-TYPE meta-entity, the relationship-type shall be sequenced using the order-clause of a relevant command.

If the RELATIONSHIP-TYPE meta-entity has meta-attribute-type SEQUENCED = NO, any meta-attribute-type SEQUENCE-PARAMETER, associated with a meta-relationship from the given RELATIONSHIP-TYPE meta-entity, whose value is YES is ignored.

SEQUENCE-PARAMETER = YES shall be specified only on one relationship-type to attribute-type meta-relationship for any given RELATIONSHIP-TYPE meta-entity. If more than one such meta-relationship exists, the RELATIONSHIP-TYPE shall not be allowed to be moved into the CONTROLLED Phase.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	N	N	N	N

**Sequenced**

**Name:** SEQUENCED  
**Description:** Identifies whether or not a relationship-type is sequenced; i.e., if it allows multiple relationships to exist between a designated pair of entities.  
**Valid Values:** YES, NO  
**Default:** NO  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	N	N	N

**Significant Attributes**

**Name:** SIGNIFICANT-ATTRIBUTES  
**Description:** Identifies the number of attribute-types which constitute the "key" for an attribute-group-type whose meta-attribute of type SINGULAR is NO.

**Valid Values:** An integer with a value of 1 to an implementation-dependent maximum value.  
**Default:** 1  
**Uniqueness Rules:** No.  
**Constraints:** If the meta-attribute associated with this meta-attribute-type exceeds the number of component-attribute-types associated with the attribute-group-type, then all component-attribute-types shall be regarded as being significant; i.e., for instances of an attribute-group of the particular attribute-group-type, the instances of corresponding significant attributes shall be unique for each group instance.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEP	N	N	N	N

**Singular**

**Name:** SINGULAR  
**Description:** Identifies whether a given attribute-type or attribute-group-type can have one or multiple values.  
**Valid Values:** YES, NO  
**Default:** YES  
**Uniqueness Rules:** No  
**Constraints:** For a given entity or relationship:  
 - Instances of a YES attribute-type or attribute-group-type can have only a single value specified.  
 - Instances of a NO attribute-type or attribute-group-type can have one to MAXIMUM-NUMBER-OF-OCCURRENCES values.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	N	N	Y

**Standard Mode**

**Name:** STANDARD-MODE  
**Description:** Identifies whether the IRDS is operating in Standard (as opposed to Extended) mode.  
**Valid Values:** YES, NO  
**Default:** YES  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	N	N	Y

**Start Name**

**Name:** START-NAME  
**Description:** A start-name identifies the initial, "starting," string of characters to be used for the assigned access name of an entity of an entity-type which has meta-attribute-type SYSTEM-GENERATED = YES.

**Valid Values:** The length of the start-name shall be less than the MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH for the corresponding entity-type.  
 A start-name is of the form  $X_1X_2...X_mN_1N_2...N_n$ , where  $X_i$ ,  $1 \leq i \leq m$ , is a non-numeric naming-character, and  $N_j$ ,  $1 \leq j \leq n$ , is a digit.

**Default:** None.

**Uniqueness Rules:** No.

**Constraints:** Applies only if entity-type's meta-attribute-type SYSTEM-GENERATED = YES.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	IDEP	N	N	N	N

(Note: The combined length of the START-NAME and the string of digits shall not exceed the value of MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH.)

### String Length Limit

**Name:** STRING-LENGTH-LIMIT

**Description:** Identifies the maximum number of characters allowed for any short-string-literal or irds-name.

**Valid Values:** An integer value not less than 72.

**Default:** 72

**Uniqueness Rules:** No.

**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	2	3	N	Y	Y	Y

### System Date

**Name:** SYSTEM-DATE

**Description:** Identifies the current date.

**Valid Values:** The digits within meta-attributes of this type conform to the mask CCYYMMDD, where:  
 CC = 2 digits, 00 through 99 inclusive, representing the century.  
 YY = 2 digits, 00 through 99 inclusive, representing the year.  
 MM = 2 digits, 01 through 12 inclusive, representing the month.  
 DD = 2 digits, 01 through 31 inclusive, representing the day of the month.

**Default:** None.

**Uniqueness Rules:** No.

**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	8	8	N	Y	N	Y

### System Generated

**Name:** SYSTEM-GENERATED

**Description:** Identifies whether or not an entity-type has system-generated entity assigned-access-names.  
**Valid Values:** YES, NO  
**Default:** NO  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	N	N	N

**System Lock**

**Name:** SYSTEM-LOCK  
**Description:** Identifies whether or not a given meta-entity or meta-relationship can be deleted from the IRD Schema by the installation.  
**Valid Values:** ON, OFF  
**Default:** OFF  
**Uniqueness Rules:** No.  
**Constraints:** This meta-attribute-type is subject to the following rules:

- (1) For meta-entities:
  - (a) The following meta-entities have SYSTEM-LOCK equal to ON:
    - The IRDS-LIMITS meta-entity, EXISTING-IRDS-LIMITS.
    - The IRDS-DEFAULTS meta-entity, EXISTING-IRDS-DEFAULTS.
    - The meta-entity of type IRDS-RESERVED-NAMES
    - Each meta-entity of type ATTRIBUTE-TYPE-VALIDATION-PROCEDURE.
    - The meta-entity of type IRD-PARTITION named SECURITY.
    - The NAMES meta-entity, NAMING-RULES.
  - (b) Any meta-entity with a NUMBER-OF-INSTANCES meta-attribute greater than zero has SYSTEM-LOCK equal to ON.
  - (c) Any meta-entity in the UNCONTROLLED IRD Schema life-cycle-phase has SYSTEM-LOCK equal to OFF.
  - (d) Any meta-entity in the CONTROLLED IRD Schema life-cycle-phase and of a type for which the modify-meta-entity-life-cycle-phase command is allowed has SYSTEM-LOCK equal to ON.
  - (e) Any other meta-entity has SYSTEM-LOCK equal to OFF.

(2) For meta-relationships:

- (a) Any meta-relationship of type:
  - RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE
  - RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-CLASS-TYPE

RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE  
 RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE  
 ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE  
 ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE  
 ENTITY-TYPE-USES-VARIATION-NAMES-DATA  
 ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE  
 ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-  
 PROCEDURE  
 ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA

in which the both meta-entities are in the CONTROLLED IRD Schema life-cycle-phase has SYSTEM-LOCK equal to ON.

(b) All other meta-relationships have SYSTEM-LOCK equal to OFF.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	Y	N	Y

**System Maintained**

Name: SYSTEM-MAINTAINED  
 Description: Identifies whether or not the assigned value(s) of the corresponding attribute-type shall be maintained by the IRDS (in which case the value is YES) or by the installation (in which case the value is NO).  
 Valid Values: YES, NO  
 Default: NO  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	N	N	N

**System Time**

Name: SYSTEM-TIME  
 Description: Identifies the current time of day.  
 Valid Values: The digits within meta-attributes of this type conform to the mask HHMMSS, where:  
 HH = 2 digits, 00 through 23 inclusive, representing the hour.  
 MM = 2 digits, 00 through 59 inclusive, representing the minute.  
 SS = 2 digits, 00 through 59 inclusive, representing the second.

(Note: Midnight is defined as HH = 00, MM = 00, SS = 00, and change of date occurs with change of SS to 01.)

Default: None.  
 Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	6	6	N	Y	N	Y

**Text In Groups Allowed**

**Name:** TEXT-IN-GROUPS-ALLOWED  
**Description:** Identifies whether or not text attributes are allowed in attribute-groups. A "YES" indicates that this extension to the standard is supported by the implementation (it does not necessarily mean that the extension has been made use of). A "NO" indicates that the implementation does not support this feature.  
**Valid Values:** YES, NO  
**Default:** None.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	2	3	N	Y	Y	Y

**Validation-Type**

**Name:** VALIDATION-TYPE  
**Description:** Identifies whether the validation of attributes of the associated attribute-type meta-entity shall be based on a list of values or a range of values.  
**Valid Values:** VALUE, RANGE  
**Default:** None.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	5	5	N	N	N	N

**Variation**

**Name:** VARIATION  
**Description:** A repeating meta-attribute-type which defines a set of variation-names which are valid for a particular entity-type.  
**Valid Values:** Any valid irds-name which does not exceed the value of VARIATION-LENGTH-LIMIT.  
**Default:** None.  
**Uniqueness Rules:** No.  
**Constraints:** None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
S	1	IDEP	Y	N	N	N

**Variation Length Limit**

**Name:** VARIATION-LENGTH-LIMIT  
**Description:** The maximum number of characters allowed for an entity variation-name.  
**Valid Values:** An integer with a value not less than 8.  
**Default:** 8

Uniqueness Rules: No.  
 Constraints: None.

Format	Minimum Length	Maximum Length	Repeating	System Maintained	Fixed	Required
N	1	IDEP	N	Y	Y	Y

**9.4 Meta-Attribute-Types for Meta-Entity Types**

Table 2 defines the correspondence between meta-entity-types and meta-attribute-types, and the rules which apply. Within this Table, and all subsequent Tables, meta-entity-types are abbreviated as follows:

- ET = ENTITY-TYPE
- RT = RELATIONSHIP-TYPE
- RCT = RELATIONSHIP-CLASS-TYPE
- AT = ATTRIBUTE-TYPE
- AGT = ATTRIBUTE-GROUP-TYPE
- ATVD = ATTRIBUTE-TYPE-VALIDATION-DATA
- ATVP = ATTRIBUTE-TYPE-VALIDATION-PROCEDURE
- IP = IRD-PARTITION
- QI = QUALITY-INDICATOR
- VND = VARIATION-NAMES-DATA
- ILMT = IRDS-LIMITS
- IDFT = IRDS-DEFAULTS
- IRSV = IRDS-RESERVED-NAMES
- NMS = NAMES

Definitions for the terms system-maintained, fixed, required, and repeating, which apply to Table 2 are given in Subsection 9.3.

**9.5 Meta-Attribute Types for Meta-Relationship Types**

Table 3 defines the correspondence between meta-relationship-types and meta-attribute-types. Note that no meta-attribute-group-types are associated with meta-relationship-types.

Definitions for the terms system-maintained, fixed, required, and repeating, which apply to the following Table are given in Subsection 9.3.

**9.6 Meta-Attribute Types for Meta-Attribute Group Types**

There exist four meta-attribute-group-types: DATA-RANGE, DATA-VALUE, DATE-TIME-ADDED, and DATE-TIME-LAST-MODIFIED.

The meta-attribute-group-types DATE-TIME-ADDED and DATE-TIME-LAST-MODIFIED are audit meta-attribute-group-types. Each is non-repeating, and consists of two component meta-attribute-types. For both of the above meta-attribute-group-types, the first meta-attribute-type is SYSTEM-DATE and the second meta-attribute-type is SYSTEM-TIME; both of these meta-attribute-types are system-maintained.

**Table 2**  
**Meta-Attribute-Type / Meta-Entity-Type Associations (Page 1 of 3)**

Meta-Attribute Type	Meta-Entity Type													
	ET	RT	RCT	AT	AGT	ATVD	ATVP	IP	QI	VND	ILMT	IDFT	IRSV	NMS
ADDED-BY	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ
COMMON-TO-ENTITY-TYPES				SF	SF									
CONNECTABLE	F													
DESCRIPTION-OF-RULES														*
ENTITY-CLASS	*													
FORMAT				Q										
IMPLEMENTATION-LOCK	F	F	F	F	F	F		F	F	F				
INTEGER-LIMIT											SFQ			
INVERSE-NAME		*	*											
LAST-MODIFIED-BY	S	S	S	S	S	S	S	S	S	S	S	S	S	S
LIFE-CYCLE-PHASE-CLASS								SF						
LINE-COUNT-LIMIT											SFQ			
LINE-LENGTH-LIMIT											SFQ			
MAXIMUM-ATTRIBUTE-LENGTH				*										
MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH	*													
MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT												Q		
MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT											SFQ			
MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH	*													
MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT												Q		
MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT											SFQ			

LEGEND: S = SYSTEM-MAINTAINED, F = FIXED, Q = REQUIRED, R = REPEATING, \* = CONTEXT-DEPENDENT



**Table 2**  
**Meta-Attribute-Type / Meta-Entity-Type Associations (Page 2 of 3)**

Meta-Attribute Type	Meta-Entity Type													
	ET	RT	RCT	AT	AGT	ATVD	ATVP	IP	QI	VND	ILMT	IDFT	IRSV	NMS
MAXIMUM-META-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT											SFQ			
MAXIMUM-META-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT											SFQ			
MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT												Q		
MAXIMUM-NUMBER-OF-OCCURRENCES-LIMIT											SFQ			
META-ENTITY-SUBSTITUTE-NAME	R	R	R	R	R	R		R	R	R				
MINIMUM-ATTRIBUTE-LENGTH				*										
MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH	*													
MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT												Q		
MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH	*													
MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT												Q		
NUMBER-OF-INSTANCES	SQ	SQ		SQ	SQ			SQ	SQ					
NUMBER-OF-TIMES-MODIFIED	S	S	S	S	S	S	S	S	S	S	S	S	S	S
ORIGIN	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ
PICTURE	R			R										
PURPOSE	*	*	*	*	*	*	*	*	*	*	*	*	*	*
RESERVED-ENTITY-NAME													SFQR	
RESERVED-META-ENTITY-NAME													SFQR	
SEQUENCED		*												
SIGNIFICANT-ATTRIBUTES					*									

LEGEND: S = SYSTEM-MAINTAINED, F = FIXED, Q = REQUIRED, R = REPEATING, \* = CONTEXT-DEPENDENT

**Table 2  
Meta-Attribute-Type / Meta-Entity-Type Associations (Page 3 of 3)**

Meta-Attribute Type	Meta-Entity Type													
	ET	RT	RCT	AT	AGT	ATVD	ATVP	IP	QI	VND	ILMT	IDFT	IRSV	NMS
STANDARD-MODE												Q		
START-NAME	*													
STRING-LENGTH-LIMIT											SFQ			
SYSTEM-GENERATED	*													
SYSTEM-LOCK	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ
SYSTEM-MAINTAINED				*										
TEXT-IN-GROUPS-ALLOWED											SFQ			
VALIDATION-TYPE						*								
VARIATION										R				
VARIATION-LENGTH-LIMIT											SFQ			

LEGEND: S = SYSTEM-MAINTAINED, F = FIXED, Q = REQUIRED, R = REPEATING, \* = CONTEXT-DEPENDENT

The meta-attribute-group-types DATA-RANGE and DATA-VALUE define mutually exclusive meta-attributes which are associated with meta-entities of type ATTRIBUTE-TYPE-VALIDATION-DATA.

DATA-RANGE defines a range of values. Each group occurrence defines an interval within the range. Both the low value and high value which define the interval are included within the interval. The intervals specified within DATA-RANGE need not be disjoint.

DATA-VALUE defines a set of values, and optionally, for each value in the set, an equivalent decoded value.

The association between these meta-attribute-group-types and their corresponding component meta-attribute-types are defined in Table 4. Position denotes the relative position of the component meta-attribute within the meta-attribute-group. Definitions for the terms system-maintained, fixed, required, and repeating, which apply to the following Table are given in Subsection 9.3.

**Table 3**  
**Meta-Attribute-Type / Meta-Relationship-Type Associations**

		Meta-Entity Type									
Meta-Entity-Type-1		RT	RT	RT	RT	ET	ET	ET	AGT	AT	AT
Meta-Relationship-Class-Type		MEMBER OF	CON-NECTS	CON-TAINS	CON-TAINS	CON-TAINS	CON-TAINS	USES	CON-TAINS	USES	USES
Meta-Entity-Type-2		RCT	ET	AGT	AT	AGT	AT	VND	AT	ATVD	ATVP
Meta-Attribute Type											
GROUP-POSITION									Q		
IMPLEMENTATION-LOCK		SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ
MAXIMUM-NUMBER-OF-OCCURRENCES				*	*	*	*				
ORIGIN		SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ	SFQ
POSITION			Q								
PURPOSE		*	*	*	*	*	*	*	*	*	*
SEQUENCE-PARAMETER					*						
SINGULAR				Q	Q	Q	Q				
SYSTEM-LOCK		SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ

LEGEND: S = SYSTEM-MAINTAINED, F = FIXED, Q = REQUIRED, R = REPEATING, \* = CONTEXT-DEPENDENT

**Table 4**  
**Meta-Attribute-Group-Type / Meta-Attribute-Type Associations**

		Meta-Attribute Group Type			
Meta-Attribute Type	Position	DATA-RANGE	DATA-VALUE	DATE-TIME-ADDED	DATE-TIME-LAST-MODIFIED
DECODED-VALUE	2		R		
ENCODED-VALUE	1		R		
HIGH-VALUE	2	*			
LOW-VALUE	1	*			
SYSTEM-DATE	1			SQ	SQ
SYSTEM-TIME	2			SQ	SQ

LEGEND: S = SYSTEM-MAINTAINED, F = FIXED, Q = REQUIRED, R = REPEATING, \* = CONTEXT-DEPENDENT

### 9.7 Meta-Attribute Group Types for Meta-Entity Types

Table 5 defines the association of meta-entity-types to meta-attribute-group-types. Definitions for the terms system-maintained, fixed, required, and repeating, which apply to the following Table are given in Subsection 9.3.

The following constraints apply for the meta-attribute-group-types in Table 5:

- 1) DATA-RANGE shall be specified only if the ATTRIBUTE-TYPE-VALIDATION-DATA meta-entity has a value of RANGE for the meta-attribute VALIDATION-TYPE.
- 2) DATA-VALUE shall be specified only if the ATTRIBUTE-TYPE-VALIDATION-DATA meta-entity has a value of VALUE for the meta-attribute VALIDATION-TYPE.

NOTE: No meta-attribute-group-type is associated with any meta-relationship-type.

### 9.8 The IRD Schema Change Control Facility

The IRDS can be viewed as having two stores of information. The principal store is the IRD. Most IRDS users use or maintain the information within this store. The second store is the IRD Schema. This store defines the logical structure of the IRD. In particular the IRD Schema defines:

- (1) The type of the entities, relationships, attributes, and attribute-groups which can be maintained in the IRD.
- (2) Basic validation rules for attribute-types and attribute-group-types.
- (3) Basic validation rules for naming entities by entity-type.
- (4) Definition of IRD-Partitions. The association of an entity with a given IRD-Partition determines constraints on modifying and deleting the entity, and relating that entity to other entities.

The IRDS provides functionality (see Subsection 5.1.1) which permits the IRD Schema to be modified. In this way, a given installation can adapt the IRDS to meet its particular requirements.

The IRD Schema is partitioned into three IRD Schema life-cycle-phases: UNCONTROLLED, CONTROLLED, and ARCHIVED. Each meta-entity exists in one and only one IRD Schema life-cycle-phase.

A meta-entity of one of the following types:

ATTRIBUTE-TYPE-VALIDATION-PROCEDURE  
NAMES  
IRDS-RESERVED-NAMES  
IRDS-LIMITS  
IRDS-DEFAULTS  
IRD-PARTITION  
QUALITY-INDICATOR

exists only in the CONTROLLED IRD Schema life-cycle-phase.

**Table 5  
Meta-Entity-type / Meta-Attribute-Type Associations**

Meta-Attribute Group Type	Meta-Entity Type													
	ET	RT	RCT	AT	AGT	ATVD	ATVP	IP	QI	VND	ILMT	IDFT	IRSV	NMS
DATA-RANGE						R								
DATA-VALUE						R								
DATE-TIME-ADDED	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ
DATE-TIME-LAST-MODIFIED	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ	SQ

LEGEND: S = SYSTEM-MAINTAINED, F = FIXED, Q = REQUIRED, R = REPEATING, \* = CONTEXT-DEPENDENT

A meta-entity of type:

ATTRIBUTE-TYPE-VALIDATION-PROCEDURE  
NAMES  
IRDS-RESERVED-NAMES  
IRDS-LIMITS

is system-created, and shall not be modified or deleted.

There shall be one IRDS-DEFAULTS meta-entity. It is also automatically created by the system. Its meta-attributes define defaults for other meta-attributes. It also shall not be deleted, but can be modified.

A meta-entity of type IRD-PARTITION or QUALITY-INDICATOR has no user modifiable meta-attributes. They shall be created in the CONTROLLED IRD Schema life-cycle-phase. They shall not be moved to any other IRD Schema life-cycle-phase.

A meta-entity of type:

RELATIONSHIP-TYPE  
RELATIONSHIP-CLASS-TYPE  
ENTITY-TYPE  
ATTRIBUTE-GROUP-TYPE  
ATTRIBUTE-TYPE  
ATTRIBUTE-TYPE-VALIDATION-DATA  
VARIATION-NAMES-DATA

may, at any one time, exist in any one of the IRD Schema life-cycle-phases. Those in the CONTROLLED IRD Schema life-cycle-phase define both the structure of the IRD and the validation of IRD entities, relationships, attributes, and attribute-groups.

An exception to the above is that the meta-entities

DATE-TIME-ADDED

DATE-TIME-ADDED  
DATE-TIME-MODIFIED  
ADDED-BY  
MODIFIED-BY  
NUMBER-OF-TIMES-MODIFIED  
SYSTEM-DATE  
SYSTEM-TIME

shall exist only in the CONTROLLED IRD Schema life-cycle-phase.

The modify meta-entity life-cycle-phase command, and the associated operating and integrity rules, provide a mechanism for controlling the evolution of the IRD structure and IRD validation parameters. A meta-entity of one of the above types shall be created in the UNCONTROLLED IRD Schema life-cycle-phase. Here they can be maintained with relative freedom. (It is anticipated that all IRD Schema maintenance is restricted.)

The functionality whereby there can exist multiple meta-entities with the same meta-entity-assigned-access-name and different version-identifiers provides a mechanism for identifying changes to IRD structure and/or validation parameters. In general, two meta-entities with the same assigned-access-name but different version-identifiers will differ in some significant fashion. This difference may be due to different meta-attributes or meta-relationships to other meta-entities.

(The relative position of meta-entity-types in meta-relationship-types can be thought of as indicating a dependency of the first meta-entity-type on the second. For example, a RELATIONSHIP-TYPE can be thought of as depending on the ENTITY-TYPE meta-entities to which it is associated. Should either of the ENTITY-TYPES be replaced by a different ENTITY-TYPE, then the RELATIONSHIP-TYPE would have been changed significantly.)

When a set of modifications to the IRD structure and/or validation parameters (e.g., VARIATION, LOW-VALUE, HIGH-VALUE, etc.) is ready to be made effective, the meta-entities are moved to the CONTROLLED IRD Schema life-cycle-phase using the modify-meta-entity-life-cycle-phase-command.

All IRD transactions reference the meta-entities in the CONTROLLED phase for the definition of IRD structure and validation rules. The modify-meta-entity-life-cycle-phase command changes these definitions. Furthermore, it may be necessary to specify multiple commands to make all the required movements of meta-entities.

If IRD accesses and/or updates were to continue while these modifications are taking place, the integrity of the IRD might be compromised and commands might not operate correctly. Thus all accesses and updates to the IRD shall be suspended while meta-entities are moved between IRD Schema life-cycle-phases. The deactivate-IRD command does this. It shall be issued prior to executing any modify-meta-entity-life-cycle-phase command, because all such commands move meta-entities either into or out of the CONTROLLED IRD Schema life-cycle-phase.

For IRD activity to be resumed, it shall be necessary that the contents of the CONTROLLED phase be compatible with IRD contents, and that no two meta-entities in the CONTROLLED phase have the same assigned-access-name. The latter constraint is called the single controlled meta-entity version constraint. It ensures that there is no ambiguity in the definition of IRD structure and validation rules.

Thus, in addition to moving new IRD structure definitions into the CONTROLLED phase, superseded meta-entities shall be moved out of the CONTROLLED phase.

By moving superseded meta-entities to the ARCHIVED phase, the contents of the ARCHIVED phase documents the prior IRD structure and validation parameters. The only maintenance allowed in the ARCHIVED IRD Schema life-cycle phase is deletion and modification of names. (The commands which modify the names of meta-entities apply to meta-entities in each IRD Schema life-cycle-phase unless a given IRD Schema life-cycle-phase is explicitly excluded.)

Once all meta-entities have been moved to the appropriate IRD Schema life-cycle-phase, IRD activity can be resumed. This is done by issuing an activate-IRD command. The activation process validates the contents of the CONTROLLED phase and enables IRD transactions. The validation includes ensuring that the IRD shall be consistent with the structure defined in the CONTROLLED phase and checking conformance with the single controlled meta-entity version constraint. The rules for IRD/IRD Schema consistency are defined in the description of the activate-IRD command.

### **9.9 The Definition of IRD Partitions in the IRD Schema**

Partitions of the IRD are defined in the IRD Schema by meta-entities of type IRD-PARTITION. Each IRD entity exists in one and only one IRD partition.

There are at least two IRD-PARTITION meta-entities. One IRD partition, with the reserved name "SECURITY", holds all entities of types required by the IRDS security facility. Entities of a type not required by the IRDS security facility shall not exist in this IRD partition. All entities of other types exist in other partitions.

Any partition other than SECURITY designates a life-cycle-phase for the IRD. (See the definition of the meta-entity-type IRD-PARTITION for details.)

IRD life-cycle-phases and IRD Schema life-cycle-phases differ in the following respects:

- (1) The IRD Schema life-cycle-phases which have been defined always exist, and no others shall exist. This is not true in the IRD. IRD life-cycle-phases can be added and deleted.
- (2) IRD life-cycle-phases are explicitly defined in the IRD Schema. IRD Schema life-cycle-phases are embedded in the functionality of IRD Schema maintenance.

As in the IRD Schema, IRD entities can be moved from one life-cycle-phase to another. The IRD command, modify-entity-life-cycle-phase, corresponds to the IRD Schema command, modify-meta-entity-life-cycle-phase. The IRD has no constraints which correspond to IRD Schema life-cycle-phase integrity or the single controlled version constraint in the IRD Schema.

## **10 The Minimal IRD Schema**

The following subsections specify the set of IRD Schema descriptors which constitute the Minimal IRD Schema.

**10.1 Purpose**

The Minimal IRD Schema consists of a set of IRD Schema descriptors. Every implementation of Module 1 of the Standard IRDS shall make available an IRD Schema which contains these descriptors.

**10.2 Minimal IRD Schema Descriptors**

An IRD Schema descriptor in the Minimal IRD Schema shall be of one of the following types:

- (1) ENTITY-TYPE
- (2) RELATIONSHIP-TYPE
- (3) RELATIONSHIP-CLASS-TYPE
- (4) ATTRIBUTE-TYPE
- (5) ATTRIBUTE-GROUP-TYPE
- (6) ATTRIBUTE-TYPE-VALIDATION-DATA
- (7) ATTRIBUTE-TYPE-VALIDATION-PROCEDURE
- (8) IRD-PARTITION
- (9) QUALITY-INDICATOR
- (10) VARIATION-NAMES-DATA
- (11) IRDS-LIMITS
- (12) IRDS-DEFAULTS
- (13) IRDS-RESERVED-NAMES
- (14) NAMES

The Minimal IRD Schema is defined by the following meta-entities, shown with their corresponding meta-entity-type:

Minimal IRD Schema Meta-Entity	Meta-Entity-Type
ADDED-BY . . . . .	. ATTRIBUTE-TYPE
ARCHIVED-LIFE-CYCLE-PHASE . . . . .	. IRD-PARTITION
CONTROLLED-LIFE-CYCLE-PHASE . . . . .	. IRD-PARTITION
DATE-TIME-ADDED . . . . .	. ATTRIBUTE-GROUP-TYPE
DATE-TIME-LAST-MODIFIED . . . . .	. ATTRIBUTE-GROUP-TYPE
DEFAULT-VIEW . . . . .	. ATTRIBUTE-TYPE
HAS . . . . .	. RELATIONSHIP-CLASS-TYPE
IRD-PARTITION-NAME . . . . .	. ATTRIBUTE-TYPE
IRDS-USER . . . . .	. ENTITY-TYPE
IRDS-USER-HAS-IRD-VIEW . . . . .	. RELATIONSHIP-TYPE
IRDS-USER-HAS-IRD-SCHEMA-VIEW . . . . .	. RELATIONSHIP-TYPE
IRD-VIEW . . . . .	. ENTITY-TYPE
EXISTING-IRDS-DEFAULTS . . . . .	. IRDS-DEFAULTS
EXISTING-IRDS-LIMITS . . . . .	. IRDS-LIMITS
LAST-MODIFIED-BY . . . . .	. ATTRIBUTE-TYPE
NAMING-RULES . . . . .	. NAMES
NUMBER-OF-TIMES-MODIFIED . . . . .	. ATTRIBUTE-TYPE
STANDARD-RESERVED-NAMES . . . . .	. IRDS-RESERVED-NAMES
RANGE-VALIDATION . . . . .	. ATTRIBUTE-TYPE-VALIDATION-PROCEDURE
IRD-SCHEMA-PHASE-NAME . . . . .	. ATTRIBUTE-TYPE
IRD-SCHEMA-VIEW . . . . .	. ENTITY-TYPE
SECURITY . . . . .	. IRD-PARTITION
SYSTEM-DATE . . . . .	. ATTRIBUTE-TYPE



SYSTEM-GENERATED-NAME . . . . . ATTRIBUTE-TYPE  
 SYSTEM-TIME . . . . . ATTRIBUTE-TYPE  
 UNCONTROLLED-LIFE-CYCLE-PHASE . . . . . IRD-PARTITION  
 VALUE-VALIDATION . . . . . ATTRIBUTE-TYPE-VALIDATION-PROCEDURE  
 YES-OR-NO-VALUE . . . . . ATTRIBUTE-TYPE-VALIDATION-DATA

**10.2.1 Entity Types in the Minimal IRD Schema**

To define the entity-types in the Minimal IRD Schema, the following information is presented for each entity-type:

- (1) The name of the entity-type, by means of the statement:

Name = <entity-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
 META-ENTITY-SUBSTITUTE-NAME  
 DATE-TIME-ADDED  
 DATE-TIME-LAST-MODIFIED  
 LAST-MODIFIED-BY  
 NUMBER-OF-TIMES-MODIFIED  
 PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

CONNECTABLE  
 ENTITY-CLASS  
 IMPLEMENTATION-LOCK  
 ORIGIN  
 SYSTEM-GENERATED  
 SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1>	...	<meta-attribute-type-name-i>	
<meta-attribute>		<meta-attribute>	

**10.2.1.1 IRDS User**

Name = IRDS-USER  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero

PURPOSE = "To identify individuals who are users of the IRDS."

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
NO	SECURITY	OFF	X3.138-1988 (Ch.1)	NO	ON

**10.2.1.2 IRD View**

Name = IRD-VIEW  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To control and regulate access to the contents of the IRD."

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
NO	SECURITY	OFF	X3.138-1988 (Ch.1)	NO	ON

**10.2.1.3 IRD Schema View**

Name = IRD-SCHEMA-VIEW  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To control and regulate access to the contents of the IRD Schema."

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
NO	SECURITY	OFF	X3.138-1988 (Ch.1)	NO	ON

**10.2.2 Relationship Types in the Minimal IRD Schema**

To define the relationship-types in the Minimal IRD Schema, the following information is presented for each relationship-type:

- (1) The name of the relationship-type, by means of the statement:

Name = <relationship-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
 META-ENTITY-SUBSTITUTE-NAME  
 DATE-TIME-ADDED  
 DATE-TIME-LAST-MODIFIED  
 INVERSE-NAME  
 LAST-MODIFIED-BY

NUMBER-OF-TIMES-MODIFIED  
PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

(3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
ORIGIN  
SEQUENCED  
SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1>	...	<meta-attribute-type-name-i>
<meta-attribute>		<meta-attribute>

**10.2.2.1 IRDS User Has IRD View**

Name = IRDS-USER-HAS-IRD-VIEW  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = IRD-VIEW-OF-IRDS-USER  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.1)	NO		ON

**10.2.2.2 IRDS User Has IRD Schema View**

Name = IRDS-USER-HAS-IRD-SCHEMA-VIEW  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = IRD-SCHEMA-VIEW-OF-IRDS-USER  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.1)	NO		ON

### 10.2.3 Relationship Class Types in the Minimal IRD Schema

To define the relationship-class-types in the Minimal IRD Schema, the following information is presented for each relationship-class-type:

- (1) The name of the relationship-class-type, by means of the statement:

Name = <relationship-class-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
DATE-TIME-ADDED  
DATE-TIME-LAST-MODIFIED  
INVERSE-NAME  
LAST-MODIFIED-BY  
NUMBER-OF-TIMES-MODIFIED  
PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
ORIGIN  
SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1>	...	<meta-attribute-type-name- <i>i</i> >
<meta-attribute>		<meta-attribute>

#### 10.2.3.1 Has

Name = HAS  
ADDED-BY = /\* implementor defined \*/  
DATE-TIME-ADDED = /\* implementor defined \*/  
DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
INVERSE-NAME = OF  
LAST-MODIFIED-BY = /\* implementor defined \*/  
NUMBER-OF-TIMES-MODIFIED = Zero  
PURPOSE = /\* implementor defined \*/

Implementation		System
Lock	Origin	Lock
ON	X3.138-1988 (Ch.1)	ON

### 10.2.4 Attribute Types in the Minimal IRD Schema

To define the attribute-types in the Minimal IRD Schema, the following information is presented for each attribute-type:

- (1) The name of the attribute-type, by means of the statement:

Name = <attribute-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
DATE-TIME-ADDED  
DATE-TIME-LAST-MODIFIED  
LAST-MODIFIED-BY  
NUMBER-OF-INSTANCES  
NUMBER-OF-TIMES-MODIFIED  
PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

COMMON-TO-ENTITY-TYPES  
FORMAT  
IMPLEMENTATION-LOCK  
ORIGIN  
SYSTEM-LOCK  
SYSTEM-MAINTAINED

by means of a table of the form:

<meta-attribute-type-name-1>	...	<meta-attribute-type-name- <i>i</i> >	
<meta-attribute>			<meta-attribute>

Attribute-types can be either constrained or unconstrained. If unconstrained, any value conforming to the specified format and lengths is valid. If an attribute-type is constrained, the valid values for the attribute-type are specified in the PURPOSE statement. If no valid values are listed in the PURPOSE statement, the attributes of the attribute-type are unconstrained.

- (4) The table

VALIDATION-PROCEDURE	VALIDATION-DATA
<meta-entity-name-1>	<meta-entity-name-2>

identifies the existence of meta-relationships of type ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-PROCEDURE and ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA. Meta-entity-name-1 and meta-entity-name-2 identify the second meta-entity in each of these meta-relationships respectively. If these meta-relationships do not exist, this table is not displayed.

An attribute-type which displays this table is said to be constrained. The valid values for the attribute-type are defined by the associated attribute-type-validation-data. If this table is not displayed, then the attribute-type is said to be unconstrained. For an unconstrained attribute-type, any value conforming to the specified length and format is valid.

**10.2.4.1 Added By**

Name = ADDED-BY  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To identify the person responsible for the creation of an instance of an IRD Schema descriptor."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock	System Maintained
YES	STRING	OFF	X3.138-1988 (Ch.1)	ON	YES

**10.2.4.2 Default View**

Name = DEFAULT-VIEW  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To designate, as an attribute-type on the IRDS-USER-HAS-IRD-VIEW or the IRDS-USER-HAS-IRD-SCHEMA-VIEW relationship-type, that a particular IRD-view or IRD-schema-view is the default view for an IRDS-USER. Valid values for this attribute-type are 'YES' and 'NO'."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock	System Maintained
NO	STRING	OFF	X3.138-1988 (Ch.1)	ON	NO

Validation-Procedure	Validation-Data
VALUE-VALIDATION	YES-OR-NO-VALUE

**10.2.4.3 IRD Partition Name**

Name = IRD-PARTITION-NAME  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To associate an IRD-Partition with an IRD-View."

Common To		Implementation		System	System
Entity-Types	Format	Lock	Origin	Lock	Maintained
NO	STRING	OFF	X3.138-1988 (Ch.1)	ON	NO

**10.2.4.4 Last Modified By**

Name = LAST-MODIFIED-BY  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To identify the person responsible for the most recent modification of the instance of an IRD Schema descriptor."

Common To		Implementation		System	System
Entity-Types	Format	Lock	Origin	Lock	Maintained
YES	STRING	OFF	X3.138-1988 (Ch.1)	ON	YES

**10.2.4.5 Number of Times Modified**

Name = NUMBER-OF-TIMES-MODIFIED  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To record the total number of times an instance of an IRD Schema descriptor has been modified."

Common To		Implementation		System	System
Entity-Types	Format	Lock	Origin	Lock	Maintained
YES	INTEGER	OFF	X3.138-1988 (Ch.1)	ON	YES

**10.2.4.6 IRD Schema Phase Name**

Name = IRD-SCHEMA-PHASE-NAME  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To associate an IRD Schema life-cycle-phase with an IRD-Schema-View."

Common To		Implementation		System	System
Entity-Types	Format	Lock	Origin	Lock	Maintained
NO	STRING	OFF	X3.138-1988 (Ch.1)	ON	NO

**10.2.4.7 System Date**

Name = SYSTEM-DATE

ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To maintain the current system date. The digits within attributes of this type conform to the mask CCYYMMDD, where:  
     CC = 2 digits, 00 through 99 inclusive, representing the century.  
     YY = 2 digits, 00 through 99 inclusive, representing the year.  
     MM = 2 digits, 01 through 12 inclusive, representing the month.  
     DD = 2 digits, 01 through 31 inclusive, representing the day of the month."

Common To	Implementation	System	System
Entity-Types	Format	Lock	Maintained
NO	DATE	OFF	YES
		Origin	
		X3.138-1988 (Ch.1)	
		System Lock	
		ON	

**10.2.4.8 System Generated Name**

Name = SYSTEM-GENERATED-NAME  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To indicate if the assigned-access-name of an entity was generated by the system."

Common To	Implementation	System	System
Entity-Types	Format	Lock	Maintained
YES	STRING	OFF	YES
		Origin	
		X3.138-1988 (Ch.1)	
		System Lock	
		ON	
	Validation-Procedure		
	VALUE-VALIDATION	Validation-Data	
		YES-OR-NO-VALUE	

**10.2.4.9 System Time**

Name = SYSTEM-TIME  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To maintain the current system time. The digits within attributes of this type conform to the mask CCYYMMDD, where:  
     HH = 2 digits, 00 through 23 inclusive, representing the hour.  
     MM = 2 digits, 00 through 59 inclusive, representing the minute.  
     SS = 2 digits, 00 through 59 inclusive, representing the second."

NOTE: Midnight is defined as HH = 00, MM = 00, SS = 00, and change of data occurs with change of SS to 01.



Common To Entity-Types	Format	Implementation Lock	Origin	System Lock	System Maintained
NO	TIME	OFF	X3.138-1988 (Ch.1)	ON	YES

### 10.2.5 Attribute Group Types in the Minimal IRD Schema

To define the attribute-group-types in the Minimal IRD Schema, the following information is presented for each attribute-group-type:

- (1) The name of the attribute-group-type, by means of the statement:

Name = <attribute-group-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
DATE-TIME-ADDED  
DATE-TIME-LAST-MODIFIED  
LAST-MODIFIED-BY  
NUMBER-OF-INSTANCES  
NUMBER-OF-TIMES-MODIFIED  
PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
ORIGIN  
SIGNIFICANT-ATTRIBUTES  
SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1>	...	<meta-attribute-type-name- <i>i</i> >
<meta-attribute>		<meta-attribute>

- (4) The table

VALIDATION-TYPE	<meta-attribute>
<meta-attribute-type-1>	<meta-attribute-type-2>
<meta-attribute(1,1)>	<meta-attribute(1,2)>
...	
<meta-attribute( <i>n</i> ,1)>	<meta-attribute( <i>n</i> ,2)>

defines the meta-attribute of type VALIDATION-TYPE and the meta-attribute-group of type DATA-RANGE or DATA-VALUE. If meta-attribute = RANGE, then the attribute-type-validation-data meta-entity has a meta-attribute-group of type DATA-RANGE. If meta-attribute = VALUE,

then the attribute-type-validation-data meta-entity has a meta-attribute-group of type DATA-VALUE. The meta-attribute-group-types in the table are the component meta-attribute-types of the meta-attribute-group-type. That is, if VALIDATION-TYPE = RANGE, then meta-attribute-type-1 = LOW-VALUE and meta-attribute-type-2 = HIGH-VALUE. If VALIDATION-TYPE = VALUE, then meta-attribute-type-1 is ENCODED-VALUE and meta-attribute-type-2 = DECODED-VALUE.

**10.2.5.1 Date Time Added**

Name = DATE-TIME-ADDED  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To specify the SYSTEM-DATE and SYSTEM-TIME an instance of an IRD Schema descriptor was created in the IRD."

Implementation		Origin	Significant Attributes	System Lock
Lock		X3.138-1988 (Ch.1)	2	ON
OFF				

**10.2.5.2 Date Time Last Modified**

Name = DATE-TIME-LAST-MODIFIED  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To specify the SYSTEM-DATE and SYSTEM-TIME an instance of an IRD Schema descriptor was last modified in the IRD."

Implementation		Origin	Significant Attributes	System Lock
Lock		X3.138-1988 (Ch.1)	2	ON
OFF				

**10.2.6 Attribute Type Validation Data in the Minimal IRD Schema**

To define the attribute-type-validation-data meta-entities in the Minimal IRD Schema, the following information is presented for each attribute-type-validation-data meta-entity:

- (1) The name of the attribute-type-validation-data meta-entity, by means of the statement:

Name = YES-OR-NO-VALUE

- (2) The meta-attributes of types:

ADDED-BY  
 DATE-TIME-ADDED

DATE-TIME-LAST-MODIFIED  
 LAST-MODIFIED-BY  
 NUMBER-OF-TIMES-MODIFIED  
 PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
 ORIGIN  
 SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1> ... <meta-attribute-type-name-i>  
 <meta-attribute>                      <meta-attribute>

**10.2.6.1 Yes Or No Value**

Name = YES-OR-NO-VALUE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To validate that an attribute is either YES or NO. "

Implementation		System
Lock	Origin	Lock
OFF	X3.138-1988 (Ch.1)	ON

Validation-Type value		Decoded-Value
Encoded-Value		YES
Y		NO
N		

**10.2.7 Attribute Type Validation Procedures in the Minimal IRD Schema**

To define the attribute-type-validation-procedures in the Minimal IRD Schema, the following information is presented for each attribute-type-validation-procedure:

- (1) The name of the attribute-type-validation-procedure, by means of the statement:

Name = <attribute-type-validation-procedure>

- (2) The meta-attributes of types:

ADDED-BY  
 DATE-TIME-ADDED  
 DATE-TIME-LAST-MODIFIED

LAST-MODIFIED-BY  
 NUMBER-OF-TIMES-MODIFIED  
 PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

(3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
 ORIGIN  
 SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1> ... <meta-attribute-type-name-i>  
 <meta-attribute>                      <meta-attribute>

**10.2.7.1 Range Validation**

Name = RANGE-VALIDATION  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To validate that an attribute is within a specified range of values."

Implementation		System
Lock	Origin	Lock
OFF	X3.138-1988 (Ch.1)	ON

**10.2.7.2 Value Validation**

Name = VALUE-VALIDATION  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To validate that an attribute equals one of a specified set of values."

Implementation		System
Lock	Origin	Lock
OFF	X3.138-1988 (Ch.1)	ON

**10.2.8 IRD Partitions in the Minimal IRD Schema**

To define the IRD-partitions in the Minimal IRD Schema, the following information is presented for each IRD-partition:

- (1) The name of the IRD-partition by means of the statement:

Name = <IRD-partition>

- (2) The meta-attributes of types:

ADDED-BY  
 DATE-TIME-ADDED  
 DATE-TIME-LAST-MODIFIED  
 LAST-MODIFIED-BY  
 NUMBER-OF-INSTANCES  
 NUMBER-OF-TIMES-MODIFIED  
 PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
 ORIGIN  
 LIFE-CYCLE-PHASE-CLASS  
 SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1> ... <meta-attribute-type-name-i>  
 <meta-attribute>                      <meta-attribute>

**10.2.8.1 Archived Life-Cycle-Phase**

Name = ARCHIVED-LIFE-CYCLE-PHASE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "The IRD-partition which is used to document those entities which are no longer in use."

Implementation		Life-Cycle	System
Lock	Origin	Phase-Class	Lock
OFF	X3.138-1988 (Ch.1)	ARCHIVED	ON

**10.2.8.2 Controlled Life-Cycle-Phase**

Name = CONTROLLED-LIFE-CYCLE-PHASE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "The IRD-partition in which entities are placed in an operational environment."

Implementation		Life-Cycle	System
Lock	Origin	Phase-Class	Lock
OFF	X3.138-1988 (Ch.1)	CONTROLLED	ON

### 10.2.8.3 Security

Name = SECURITY  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "The IRD-partition which is used for those entities of type IRDS-USER."

Implementation		Life-Cycle	System
Lock	Origin	Phase-Class	Lock
OFF	X3.138-1988 (Ch.1)		ON

### 10.2.8.4 Uncontrolled Life-Cycle-Phase

Name = UNCONTROLLED-LIFE-CYCLE-PHASE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "The IRD-partition in which entities can be added and modified."

Implementation		Life-Cycle	System
Lock	Origin	Phase-Class	Lock
OFF	X3.138-1988 (Ch.1)	UNCONTROLLED	ON

## 10.2.9 IRDS Defaults in the Minimal IRD Schema

To define the IRDS-DEFAULTS meta-entity in the Minimal IRD Schema, the following information is presented:

- (1) The name of the IRDS-DEFAULTS meta-entity by means of the statement:

Name = EXISTING-IRDS-DEFAULTS

- (2) The meta-attributes of the meta-attribute-types applicable to meta-entities of type IRDS-defaults by means of the form:

<meta-attribute-type-name> = <meta-attribute>

### 10.2.9.1 Existing IRDS Defaults

Name = EXISTING-IRDS-DEFAULTS  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 IMPLEMENTATION-LOCK = OFF

```

MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT =
/* implementor defined */
MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT =
/* implementor defined */
MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT = /* implementor defined */
MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-DEFAULT =
/* implementor defined */
MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-DEFAULT =
/* implementor defined */
ORIGIN = X3.138-1988 (Ch.1)
STANDARD-MODE = /* implementor defined */
SYSTEM-LOCK = ON

```

### 10.2.10 IRDS Limits in the Minimal IRD Schema

To define the IRDS-LIMITS meta-entity in the Minimal IRD Schema, the following information is presented:

- (1) The name of the IRDS-LIMITS meta-entity by means of the statement:

```
Name = EXISTING-IRDS-LIMITS
```

- (2) The meta-attributes of the meta-attribute-types applicable to meta-entities of type IRDS-limits by means of the form:

```
<meta-attribute-type-name> = <meta-attribute>
```

#### 10.2.10.1 Existing IRDS Limits

```

Name = EXISTING-IRDS-LIMITS
ADDED-BY = /* implementor defined */
DATE-TIME-ADDED = /* implementor defined */
DATE-TIME-LAST-MODIFIED = /* implementor defined */
IMPLEMENTATION-LOCK = OFF
INTEGER-LIMIT = /* implementor defined */
LAST-MODIFIED-BY = /* implementor defined */
LINE-COUNT-LIMIT = /* implementor defined */
LINE-LENGTH-LIMIT = /* implementor defined */
MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT = /* implementor defined */
MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT =
/* implementor defined */
MAXIMUM-META-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-LIMIT =
/* implementor defined */
MAXIMUM-META-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-LENGTH-LIMIT =
/* implementor defined */
MAXIMUM-NUMBER-OF-OCCURRENCES-LIMIT = /* implementor defined */
NUMBER-OF-TIMES-MODIFIED = Zero
ORIGIN = X3.138-1988 (Ch.1)
STRING-LENGTH-LIMIT = /* implementor defined */
SYSTEM-LOCK = ON
VARIATION-LENGTH-LIMIT = /* implementor defined */

```

### 10.2.11 IRDS Reserved Names in the Minimal IRD Schema

To define the IRDS-RESERVED-NAMES meta-entity in the Minimal IRD Schema, the following information is presented:

- (1) The name of the IRDS-RESERVED-NAMES meta-entity by means of the statement:

Name = STANDARD-RESERVED-NAMES

- (2) The meta-attributes of the meta-attribute-types applicable to meta-entities of type IRDS-Reserved-Names by means of the form:

<meta-attribute-type-name> = <meta-attribute>

#### 10.2.11.1 Standard Reserved Names

Name = STANDARD-RESERVED-NAMES  
ADDED-BY = /\* implementor defined \*/  
DATE-TIME-ADDED = /\* implementor defined \*/  
IMPLEMENTATION-LOCK = OFF  
ORIGIN = X3.138-1988 (Ch.1)  
SYSTEM-LOCK = ON

### 10.2.12 Names in the Minimal IRD Schema

To define the NAMES meta-entity in the Minimal IRD Schema, the following information is presented:

- (1) The name of the NAMES meta-entity by means of the statement:

Name = NAMING-RULES

- (2) The meta-attributes of the meta-attribute-types applicable to meta-entities of type Names by means of the form:

<meta-attribute-type-name> = <meta-attribute>

#### 10.2.12.1 Naming Rules

Name = NAMING-RULES  
ADDED-BY = /\* implementor defined \*/  
DATE-TIME-ADDED = /\* implementor defined \*/  
DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
IMPLEMENTATION-LOCK = OFF  
LAST-MODIFIED-BY = /\* implementor defined \*/  
NUMBER-OF-TIMES-MODIFIED = Zero  
ORIGIN = X3.138-1988 (Ch.1)  
SYSTEM-LOCK = ON  
DESCRIPTION-OF-RULES =  
/\* Subsection 4.4, using wording determined by the implementation, shall be inserted here.  
\*/



### 10.3 Meta-Relationships in the Minimal IRD Schema

The associations between these descriptors are expressed in terms of the meta-relationships that exist in the IRD Schema. Applicable meta-attributes are specified.

#### 10.3.1 Relationship Type Connects Entity Type

There are two relationship-types found in the Minimal IRD Schema, which are associated with entity-types as follows:

- (1) The IRDS-USER-HAS-IRD-VIEW relationship-type has the IRDS-USER entity-type as the first member and the IRD-VIEW entity-type as the second member.
- (2) The IRDS-USER-HAS-IRD-SCHEMA-VIEW relationship-type has the IRDS-USER entity-type as the first member and the IRD-SCHEMA-VIEW entity-type as the second member.

#### 10.3.2 Relationship Type Member of Relationship Class Type

The Minimal IRD Schema shall contain the following meta-relationships of type RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-CLASS-TYPE:

The relationship-types:

IRDS-USER-HAS-IRD-VIEW and  
IRDS-USER-HAS-IRD-SCHEMA-VIEW

and the relationship-class-type HAS.

#### 10.3.3 Entity Type Contains Attribute Type

The Minimal IRD Schema shall contain meta-relationships of the type ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE according to Table 6.

#### 10.3.4 Entity Type Contains Attribute Group Type

The Minimal IRD Schema shall contain meta-relationships of type ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE according to Table 7.

#### 10.3.5 Relationship Type Contains Attribute Type

The Minimal IRD Schema shall contain meta-relationships of type RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE as follows:

The relationship-types IRDS-USER-HAS-IRD-VIEW and IRDS-USER-HAS-IRD-SCHEMA-VIEW are associated with the attribute-type DEFAULT-VIEW, where the meta-entity of type SINGULAR is YES.

#### 10.3.6 Relationship Type Contains Attribute Group Type

The Minimal IRD Schema does not contain any meta-relationships of type RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE.

**Table 6**  
**Entity-Type / Attribute-Type Associations**

Attribute Type	Entity Type		
	IUSER	IVW	ISVW
ADDED-BY	S	S	S
IRD-PARTITION-NAME		S	
LAST-MODIFIED-BY	S	S	S
NUMBER-OF-TIMES-MODIFIED	S	S	S
IRD-SCHEMA-PHASE-NAME			S

LEGEND: S = SINGULAR, P = PLURAL

**Table 7**  
**Entity-Type / Attribute-Group-Type Associations**

Attribute Group Type	Entity Type		
	IUSER	IVW	ISVW
DATE-TIME-ADDED	S	S	S
DATE-TIME-LAST-MODIFIED	S	S	S

LEGEND: S = SINGULAR, P = PLURAL

**10.3.7 Attribute Group Type Contains Attribute Type**

The Minimal IRD Schema shall contain meta-relationships of type ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE as follows:

For the attribute-group-type DATE-TIME-ADDED:

The attribute-type SYSTEM-DATE with meta-attribute of type GROUP-POSITION equal to 1;

The attribute-type SYSTEM-TIME with meta-attribute of type GROUP-POSITION equal to 2.

For the attribute-group-type DATE-TIME-LAST-MODIFIED:

The attribute-type SYSTEM-DATE with meta-attribute of type GROUP-POSITION equal to 1;

The attribute-type SYSTEM-TIME with meta-attribute of type GROUP-POSITION equal to 2.

### 10.3.8 Attribute Type Uses Attribute Type Validation Data

The Minimal IRD Schema shall contain a meta-relationship of type ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-DATA as follows:

For the attribute-type DEFAULT-VIEW, the attribute-type-validation-data YES-OR-NO-VALUE.

The attribute of type IRD-SCHEMA-PHASE-NAME equal to CONTROLLED.

### 10.3.9 Relationships and Their Attributes

The following relationships and associated attributes shall exist in the IRD:

A relationship of type IRDS-USER-HAS-IRD-VIEW with members ADMINISTRATOR and ADMINISTRATOR-IRD-VIEW, with the following attribute:

The attribute of type DEFAULT-VIEW equal to YES.

A relationship of type IRDS-USER-HAS-IRD-SCHEMA-VIEW with members ADMINISTRATOR and ADMINISTRATOR-IRD-SCHEMA-VIEW, with the following attribute:

The attribute of type DEFAULT-VIEW equal to YES.

## 11 The Information Resource Dictionary (IRD)

This section specifies the Information Resource Dictionary (IRD) which shall be made available within an implementation of Module 1 of the standard IRDS. The information presented here is the same as that found in Subsection 5.3.1.1, but is repeated for ease of reference.

### 11.1 Entities and Their Attributes

The following entities and associated attributes shall exist in the IRD:

An entity of type IRDS-USER with assigned-access-name ADMINISTRATOR.

An entity of type IRD-VIEW with assigned-access-name ADMINISTRATOR-IRD-VIEW, with the following attribute:

The attribute of type IRD-PARTITION-NAME equal to SECURITY.

An entity of type IRD-SCHEMA-VIEW with assigned-access-name ADMINISTRATOR-IRD-SCHEMA-VIEW, with the following attribute:

The attribute of type IRD-SCHEMA-PHASE-NAME equal to CONTROLLED.

## 11.2 Relationships and Their Attributes

The following relationships and associated attributes shall exist in the IRD:

A relationship of type IRDS-USER-HAS-IRD-VIEW with members ADMINISTRATOR and ADMINISTRATOR-IRD-VIEW, with the following attribute:

The attribute of type DEFAULT-VIEW equal to YES.

A relationship of type IRDS-USER-HAS-IRD-SCHEMA-VIEW with members ADMINISTRATOR and ADMINISTRATOR-IRD-SCHEMA-VIEW, with the following attribute:

The attribute of type DEFAULT-VIEW equal to YES.

# Appendix

(This Appendix is not part of American National Standard X3.138-1988, but is included for information only.)

## Collected Syntax

```
access-name ::=
    assigned-access-name
    [ version-identifier ]

access-name-component ::=
    assigned-word access-name-word
    | version-word
    | variation-word
    | revision-word

access-name-component-list ::=
    access-name-component
    [ , access-name-component-list ]

access-name-display-option ::=
    access-name-word
    | access-name-component-list

access-name-pattern ::=
    name-scan-pattern
    | entity-access-name

access-name-scan-pattern-list ::=
    access-name-pattern
    [ , access-name-scan-pattern-list ]

activate-IRD-command ::=
    activate-word
    IRD-word
    [ implementation-defined-options ]
    ;

add-entity-command ::=
    add-word
    entity-word
    new-entity-name
    entity-type-clause
    [ entity-descriptive-name-declaration-clause ]
    [ quality-indicator-designation-clause ]
    [ new-entity-attributes-clause ]
    ;

add-meta-entity-command ::=
    add-word
    meta-entity-word
    new-meta-entity-access-name
    meta-entity-type-clause
    [ meta-entity-descriptive-name-clause ]
    [ new-meta-attributes-clause ]
    ;

add-meta-relationship-command ::=
    add-word
    meta-relationship-word
    meta-relationship-identifier
    [ new-meta-attributes-clause ]
    ;

add-relationship-command ::=
    add-word
    relationship-word
    new-relationship-identification-clause
    [ new-relationship-attributes-clause ]
    ;

after-last-phrase ::=
    after-word
    [ last-word
    [ entity-word ] ]

all-attributes-except-phrase ::=
    all-word
    attributes-word
    except-word
    attribute-selection-list

all-attributes-phrase ::=
    all-word
    attributes-word

all-entities-clause ::=
    all-word
    [ entities-word ]
```

## APPENDIX

```

all meta-attributes-option ::=
    all-word
    meta-attributes-word

all-option ::=
    all-word
    entities-word

all-or-no-option ::=
    all-word
    | no-word

all-or-no-word ::=
    all-word
    | no-word

all-or-specified-option ::=
    all-meta-attributes-option
    | specified-meta-attributes-option

all-or-status-options ::=
    all-word
    | status-option-list

all-relationships-option ::=
    all-word
    [ forward-or-inverse-option ]
    relationships-word

all-word-or-new-word ::=
    all-word
    | new-word

alternate-name ::=
    irds-name
    | name-scan-mask

alternate-name-restriction-clause ::=
    alternate-name-word
    =
    alternate-name
    [ in-word
    context-word
    context ]

ascending-or-descending ::=
    ascending-word
    | descending-word

ascending-or-descending-option ::=
    ascending-word
    | descending-word

assigned-access-name ::= irds-name

assigned-descriptive-name ::= irds-name

asterisk ::= *

attribute ::=
    text-attribute
    | non-text-attribute

attribute-1 ::= non-text-attribute

attribute-decoding-option ::=
    show-word
    attributes-word
    encoded-or-decoded

attribute-display-option ::=
    and-word
    [ all-or-no-option ]
    attributes-word

attribute-group ::=
    attribute-group-positional-format
    | attribute-group-non-positional-format

attribute-group-list ::=
    attribute-group
    [ [,] attribute-group-list ]

attribute-group-non-positional-format ::=
    ( component-attribute-type-1-designator =
    component-attribute-1
    { [,] component-attribute-type-i-designator =
    component-attribute-i } )

attribute-group-phrase ::=
    attribute-group-type-designator
    [ (component-attribute-type-list) ]

attribute-group-positional-format ::=
    ( attribute-1 | null
    {, attribute-i | null }
    [, last-attribute-of-group ] )

attribute-group-restriction-clause ::=
    for-word
    attribute-group-type-designator
    component-attribute-restriction-expression

attribute-group-type-designator ::=
    meta-entity-designator

attribute-i ::= non-text-attribute

```

```

attribute-occurrence-qualification ::=
    ( attribute )

attribute-restriction-clause ::=
    attribute-type-designator
    relational-operator
    [ only-word ]
    test-value-list

attribute-selection-list ::=
    selected-attribute-or-attribute-group
    [ ,attribute-selection-list ]

attribute-selection-option ::=
    all-attributes-phrase
    | no-attributes-phrase
    | only-specified-attributes-phrase
    | specified-attributes-phrase
    | all-attributes-except-phrase

attribute-type-designator ::=
    meta-entity-designator

audit-attribute-restriction-clause ::=
    audit-attribute-type
    relational-operator
    test-value

audit-attribute-type ::=
    attribute-type-designator

audit-meta-attribute-group-type ::=
    DATE-TIME-ADDED
    | DATE-TIME-LAST-MODIFIED

audit-meta-attribute-type ::=
    ADDED-BY
    | LAST-MODIFIED-BY
    | NUMBER-OF-TIMES-MODIFIED

boolean-operator ::=
    and-word | or-word | xor-word | not-word

century-code ::= digit
    digit

character ::=
    letter
    | digit
    | special-character

```

```

check-IRD-schema-compatibility-command ::=
    check-word
    IRD-schema-word
    other-IRD-schema-clause
    [ implementor-defined-clauses ]
    ;

command-imperative ::=
/* The command imperative is the sequence of irds-
words which:
    begin a command,
    uniquely identify that command, and
    are not part of any command-clause.

For example, the command-imperative for the add-
entity-command consists of the two irds-words in
order: add-word entity-word.

The concept is important for identifying a command
and identifying where the next command is should
the prior command be invalid and not properly
terminated. */

command-imperative-substring ::=
/* A command-imperative-substring is a sequence of
irds-words which are used in IRDS command-
imperatives. */

command-imperative-substring-option ::=
    command-imperative-word
    [ command-imperative-substring ]

command-imperative-word ::= COMMAND-IMPERATIVE

comment ::=
    slash asterisk {character}
    asterisk slash

comparison-value ::=
    unsigned-integer
    | lowest-word
    | highest-word

component-attribute-1 ::=
    non-text-attribute

component-attribute-1-designator ::=
    irds-name

component-attribute-i ::=
    non-text-attribute

```

## APPENDIX

```

component-attribute-restriction-expression ::=
    attribute-restriction-clause
    | ( component-attribute-restriction-expression )
    | component-attribute-restriction-expression
    | boolean-operator
    | component-attribute-restriction-expression

component-attribute-type-i-designator ::=
    irds-name

component-meta-attribute-restriction ::=
    component-meta-attribute-type
    | relational-operator
    | meta-attribute

component-meta-attribute-type ::=
    DECODED-VALUE
    | ENCODED-VALUE
    | HIGH-VALUE
    | LOW-VALUE
    | SYSTEM-DATE
    | SYSTEM-TIME

component-restriction-expression ::=
    [ component-restriction-expression
      | boolean-operator
      | component-restriction-expression ]
    | component-restriction-expression
    | component-meta-attribute-restriction

context ::=
    irds-name
    | short-string-literal
    | string-scan-mask

control-meta-attribute-type ::=
    IMPLEMENTATION-LOCK
    | LIFE-CYCLE-PHASE-CLASS
    | NUMBER-OF-INSTANCES
    | SYSTEM-LOCK
    | SYSTEM-MAINTAINED

controlled-only-option ::=
    controlled-word
    | only-word

copy-entity-command ::=
    copy-word
    | entity-word
    | existing-entity-access-name
    | [ with-relationships-clause ]
    | to-word
    | new-entity
    | [ entity-descriptive-name-declaration-clause ]
    | [ quality-indicator-designation-clause ]
    ;

copy-meta-entity-command ::=
    copy-word
    | meta-entity-word
    | existing-meta-entity-access-name
    | [ with-meta-relationships-clause ]
    | to-word
    | new-meta-entity-specification
    | [ meta-entity-descriptive-name-clause ]
    ;

count-function ::=
    count-function-qualifier'COUNT

count-function-qualifier ::=
    attribute-type-designator
    | meta-attribute-type
    | attribute-group-type-designator
    | meta-attribute-group-type

create-IRD-command ::=
    create-word
    | IRD-word
    | new-IRD-name
    | [ location-clause ]
    | IRD-schema-source-clause
    | [ load-IRD-clause ]
    | [ implementor-defined-clauses ]
    ;

cumulative-impact-format ::=
    cumulative-word
    | impact-word
    | impact-show-options

current-IRD-schema-life-cycle-phase ::=
    IRD-schema-life-cycle-phase

current-life-cycle-phase ::=
    irds-name

date-literal ::=
    specified-date
    | current-date-word

```



```

date-time-literal ::=
    date-literal
    | time-literal

day-code ::=
    01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16
    |17|18|19|20|21|22|23|24|25|26|27|28|29|30|31

deactivate-IRD-command ::=
    deactivate-word
    IRD-word
    [ implementation-defined-options ]
    ;

decimal-point-character ::= .

delete-entity-command ::=
    delete-word
    entity-word
    delete-selection-option
    [ with-relationships-clause ]
    ;

delete-meta-entity-command ::=
    delete-word
    meta-entity-word
    existing-meta-entity-access-name
    [ with-meta-relationships-clause ]
    ;

delete-meta-relationship-command ::=
    delete-word
    meta-relationship-word
    meta-relationship-identifier
    ;

delete-relationship-command ::=
    delete-word
    relationship-word
    relationships-identification
    ;

delete-selection-option ::=
    entity-name-list
    | entity-selection-criteria-clause

delimiter-token ::=
    string-literal
    | relational-operator
    | : | ; | ( | ) | ,

descriptive-name ::=
    assigned-descriptive-name
    [ version-identifier ]

descriptive-name-component ::=
    assigned-word descriptive-name-word
    | version-word
    | variation-word
    | revision-word

descriptive-name-component-list ::=
    descriptive-name-component
    [ , descriptive-name-component-list ]

descriptive-name-display-option ::=
    descriptive-name-word
    | descriptive-name-component-list

descriptive-name-pattern ::=
    name-scan-pattern
    | entity-descriptive-name

descriptive-name-scan-pattern-list ::=
    descriptive-name-pattern
    [ , descriptive-name-scan-pattern-list ]

destination ::=
    /* implementor-defined-format */

destination-list ::=
    destination
    [ , destination-list ]

digit ::=
    0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

directly-or-indirectly-option ::=
    directly-word
    | indirectly-word

display-format ::=
    standard-display-format-option
    | implementor-defined-format-option

each-page-option ::=
    on-word
    each-word
    page-word

effective-view-option ::=
    IRD-word | IRD-schema-word
    view-word
    =
    security-entity-name

```

APPENDIX

```

encoded-or-decoded ::=
    encoded-word
    | decoded-word

enter-panel-dialogue-command ::=
    panel-word
    [ panel-transfer-option ]
    ;

entity-1-name ::=
    entity-access-name

entity-2-entity-type-designator ::=
    entity-type-designator

entity-2-name ::=
    entity-access-name

entity-2-name-or-null-mark ::=
    entity-2-name
    | null-mark

entity-access-name ::=
    new-entity-access-name
    | existing-entity-access-name

entity-access-name-list ::=
    entity-access-name
    [ , entity-access-name-list ]

entity-access-name-phrase ::=
    [ assigned-word ]
    access-name-word

entity-access-name-selection-clause ::=
    entities-word
    [ with-word ]
    access-name-word
    =
    access-name-scan-pattern-list

entity-access-names-or-list ::=
    entity-access-name-list

entity-and-relationship-type-qualified-shows ::=
    show-clause-options
    relationship-type-qualified-shows
    { relationship-type-qualified-shows }

entity-assigned-access-name-restriction-clause ::=
    entity-word
    assigned-word
    access-name-word
    relational-operator
    irds-name

entity-assigned-descriptive-name-
restriction-clause ::=
    entity-word
    assigned-word
    descriptive-name-word
    relational-operator
    irds-name

entity-descriptive-name ::=
    descriptive-name

entity-descriptive-name-declaration-clause ::=
    entity-word
    descriptive-name-word
    = assigned-descriptive-name

entity-descriptive-name-phrase ::=
    [ assigned-word ]
    descriptive-name-word

entity-descriptive-name-selection-clause ::=
    entities-word
    [ with-word ]
    descriptive-name-word
    =
    descriptive-name-scan-pattern-list

entity-name-list ::=
    entity-access-name
    [ , entity-name-list ]

entity-selection ::=
    all-entities-clause
    | entity-access-name-selection-clause
    | entity-descriptive-name-selection-clause
    | related-entities-clause

entity-selection-criteria-clause ::=
    select-word
    entity-selection
    [ where-word
    restriction-expression ]

entity-type-clause ::=
    entity-type-word
    =
    entity-type-designator

```

```

entity-type-designator ::=
    meta-entity-designator

entity-type-list ::=
    entity-type-designator
    [ , entity-type-list ]

entity-type-option ::=
    all-option
    | only-entity-types-option

entity-type-qualification-clause ::=
    for-word
    [ entities-word
      of-word ]
    entity-type-list

entity-type-restriction-clause ::=
    entity-type-word
    =
    entity-type-list

entity-type-show-restriction-clause ::=
    show-word
    entity-type-option

equals ::= =

equals-or-not-equal ::=
    equals
    | not-equal

exclude-relationships-of-type-clause ::=
    exclude-word
    [ relationships-word
      of-word ]
    relationship-type-or-class-type-list

excluding-relationships-option ::=
    all-word
    relationships-word
    except-word
    relationship-type-selection-list

existing-entity-access-name ::=
    access-name

existing-entity-assigned-access-name ::=
    assigned-access-name

existing-entity-assigned-descriptive-name ::=
    assigned-descriptive-name

existing-entity-specification ::=
    using-entity-assigned-descriptive-name-
        specification
    | using-entity-assigned-access-name-
        specification

existing-group-occurrence ::=
    meta-attribute-group

existing-meta-entity-access-name ::=
    access-name

existing-meta-entity-assigned-access-name ::=
    assigned-access-name

existing-meta-entity-assigned-descriptive-name ::=
    assigned-descriptive-name

existing-meta-entity-specification ::=
    using-meta-entity-assigned-descriptive-name-
        specification
    | using-meta-entity-assigned-access-name-
        specification

exit-IRDS-system-command ::=
    exit-word
    ;

exponent-character ::= e | E

export-IRD-command ::=
    export-word
    IRD-word
    [ using-IRD-views-clause ]
    extraction-option
    [ exclude-relationship-of-type-clause ]
    IRD-schema-export-file-clause
    [ controlled-only-option ]
    [ IRD-export-file-clause ]
    [ file-title-suffix-clause ]
    [ implementor-defined-clauses ]
    ;

extraction-option ::=
    | IRD-schema-only-option

file-name ::=
    /* implementor-defined-format */

file-title-suffix-clause ::=
    suffix-word
    =
    short-string-literal

```

APPENDIX

```

first-default-clause ::=
    start-line-number-clause
    | line-number-increment-clause
  
```

```

first-or-each-page-option
    first-page-option
    | each-page-option
  
```

```

first-page-option ::=
    on-word
    first-word
    page-word
  
```

```

fixed-point-literal ::=
    integer [.unsigned-integer]
    | integer.
    | [ + | - ] .unsigned-integer
  
```

```

for-each-phrase ::=
    for-word
    each-word
    [ entity-word ]
  
```

```

forward-association-designator ::=
    relationship-type-designator
    | relationship-class-type-designator
  
```

```

forward-inverse-criterion ::=
    forward-word
    | inverse-word
  
```

```

forward-or-inverse-option ::=
    forward-word
    | inverse-word
  
```

```

forward-or-inverse-word ::=
    forward-word
    | inverse-word
  
```

```

fractional-part ::= unsigned-integer
  
```

```

fractional-real ::=
    unsigned-integer
    decimal-point-character
    fractional-part
    [ exponent-character
      scale-factor ]
  
```

```

function-result ::=
    count-function-result
    | lines-function-result
  
```

```

general-command ::=
    ird-ird-interface-commands
    | utility-commands
  
```

```

general-output-command ::=
    output-word
    IRD-word
    [ using-IRD-views-clause ]
    selection-option
    [ sort-clause ]
    show-options
    [ route-clause ]
    [ implementor-defined-clauses ]
    ;
  
```

```

general-output-show-clause-option ::=
    show-all-clause
    | show-entity-type-clause
    | show-IRD-life-cycle-phase-clause
    | show-entity-access-name-clause
    | show-entity-descriptive-name-clause
    | show-relationships-clause
    | show-attributes-clause
    | show-counts-clause
    | show-quality-indicator-clause
  
```

```

greater-than ::= >
  
```

```

greater-than-or-equal ::= >=
  
```

```

group-occurrence-qualification ::=
    attribute-group
  
```

```

help-command ::=
    help-word
    [ help-option ]
    ;
  
```

```

help-option ::=
    message-option
    | all-word
    | command-imperative-substring-option
  
```

```

hour-code ::=
    00|01|02|03|04|05|06|07|08|09|10|11
    |12|13|14|15|16|17|18|19|20|21|22|23
  
```

```

impacted-entity-show-clause ::=
    show-IRD-life-cycle-phase-clause
    show-quality-indicator-clause
    show-attributes-clause
    show-entity-descriptive-name-clause
  
```

```

impacted-entity-show-clause-list ::=
    impacted-entity-show-clause
    [ impacted-entity-show-clause-list ]

impact-option ::=
    cumulative-word
    | individual-word

impact-show-options ::=
    [ show-title-clause ]
    [ entity-type-show-restriction-clause ]
    { impacted-entity-show-clause }
    { qualified-impacted-entity-show-clause }

implementor-defined-clauses ::=
    /* implementor-defined-format */

implementor-defined-format-option ::=
    /* implementor-defined-format */

implementor-defined-option ::=
    /* implementor-defined-format */

import-IRD-command ::=
    import-word
    IRD-word
    IRD-schema-export-file-clause
    IRD-export-file-clause
    IRD-life-cycle-phase-designation-clause
    [ implementor-defined-clauses ]
    ;

in-IRD-clause ::=
    in-word
    IRD-word
    IRD-name
    [ location-clause ]

in-file-clause ::=
    in-word
    file-word
    file-name
    [ location-clause ]

increment ::= unsigned-integer

increment-value ::= unsigned-integer

individual-impact-format ::=
    [ individual-word ]
    impact-word
    impact-show-options

integer-literal ::= integer

integer ::=
    signed-integer
    | unsigned-integer

inverse-association-designator ::=
    relationship-type-inverse-name
    | relationship-class-type-inverse-name

inverse-name ::=
    irds-name

IRD-command ::=
    IRD-maintenance-command
    | IRD-output-command

IRD-export-file-clause ::=
    IRD-word
    export-word
    file-word
    =
    IRD-export-file-name
    [ location-clause ]

IRD-export-file-name ::=
    /* implementor-defined-format */

IRD-IRD-interface-command ::=
    create-IRD-command
    | export-IRD-command
    | check-IRD-schema-compatibility-command
    | import-IRD-command

IRD-life-cycle-phase-designation-clause ::=
    IRD-word
    life-cycle-phase-word
    =
    IRD-life-cycle-phase-designator

IRD-life-cycle-phase-designator ::=
    meta-entity-designator

IRD-life-cycle-phase-list ::=
    IRD-life-cycle-phase-designator
    [ , IRD-life-cycle-phase-list ]

IRD-life-cycle-phase-restriction-clause ::=
    IRD-word
    life-cycle-phase-word
    relational-operator
    IRD-life-cycle-phase-designator

```

APPENDIX

```

IRD-maintenance-command ::=
    add-entity-command
    | modify-entity-command
    | delete-entity-command
    | add-relationship-command
    | modify-relationship-command
    | delete-relationship-command
    | modify-entity-access-name-command
    | modify-entity-descriptive-name-command
    | modify-entity-life-cycle-phase-command
    | copy-entity-command

IRD-name ::=
    /* implementor-defined-format */

IRD-name-space ::=
    { entity-access-name }
    { entity-descriptive-name }

IRD-output-commands ::=
    general-output-command
    | output-impact-of-change-command
    | [ output-syntax-command ]

IRD-schema-command ::=
    IRD-schema-maintenance-command
    | IRD-schema-output-command

IRD-schema-descriptor-qualification ::=
    meta-entity-type-qualification-clause
    | meta-relationship-type-qualification-clause

IRD-schema-export-file-clause ::=
    IRD-schema-word
    export-word
    file-word
    =
    IRD-schema-export-file-name
    [ location-clause ]

IRD-schema-export-file-name ::=
    /* implementor-defined-format */

IRD-schema-life-cycle-phase ::=
    UNCONTROLLED
    | CONTROLLED
    | ARCHIVED

IRD-schema-life-cycle-phase-restriction-clause ::=
    IRD-schema-word
    life-cycle-phase-word
    relational-operator
    IRD-schema-life-cycle-phase

```

```

IRD-schema-maintenance-command ::=
    add-meta-entity-command
    | modify-meta-entity-command
    | delete-meta-entity-command
    | add-meta-relationship-command
    | modify-meta-relationship-command
    | delete-meta-relationship-command
    | modify-meta-entity-access-name-command
    | modify-meta-entity-descriptive-name-command
    | modify-meta-entity-life-cycle-phase-command
    | copy-meta-entity-command
    | deactivate-IRD-command
    | restore-IRD-schema-command
    | activate-IRD-command

IRD-schema-name-space ::=
    { meta-entity-access-name }
    { meta-entity-descriptive-name }

IRD-schema-only-option ::=
    IRD-schema-word
    only-word

IRD-schema-output-command ::=
    output-word
    IRD-schema-word
    [ using-IRD-schema-views-clause ]
    meta-entity-selection-clause
    [ sort-meta-entities-clause ]
    show-IRD-schema-options
    [ route-clause ]
    ;

IRD-schema-restriction-clause ::=
    meta-entity-type-restriction-clause
    | meta-relationship-existence-restriction-clause
    | meta-attribute-group-restriction-clause
    | meta-attribute-restriction-clause
    | IRD-schema-life-cycle-phase-restriction-clause

IRD-schema-restriction-expression ::=
    IRD-schema-restriction-expression
    boolean-operator
    IRD-schema-restriction-expression
    ( IRD-schema-restriction-expression )
    [ IRD-schema-descriptor-qualification ]
    IRD-schema-restriction-expression
    | IRD-schema-restriction-clause

IRD-schema-show-all-clause ::=
    show-word
    all-word

```

```

IRD-schema-show-clause-list ::=
    IRD-schema-show-clause-option
    [ IRD-schema-show-clause-list ]

IRD-schema-show-clause-option ::=
    IRD-schema-show-all-clause
    | show-meta-attributes-clause
    | show-meta-relationships-clause
    | show-related-meta-entities-clause
    | show-IRD-schema-life-cycle-phase-clause

IRD-schema-sort-parameter ::=
    IRD-schema-sort-parameter-form-1
    IRD-schema-sort-parameter-form-2

IRD-schema-sort-parameter-form-1 ::=
    meta-entity-type-word
    | simple-meta-attribute-type
    | life-cycle-phase-word

IRD-schema-sort-parameter-form-2 ::=
    ( IRD-schema-sort-parameter-form-1
      , ascending-or-descending-option )

IRD-schema-sort-parameter-list ::=
    IRD-schema-sort-parameter
    [ , IRD-schema-sort-parameter-list ]

IRD-schema-source-clause ::=
    IRD-schema-word
    [ is-word ]
    IRD-schema-source-option

IRD-schema-source-option ::=
    in-file-clause
    | in-IRD-clause
    | minimal-word

IRD-schema-view-list ::=
    IRD-schema-view-name
    [ , IRD-schema-view-list ]

IRD-schema-view-name ::=
    assigned-access-name

IRD-sort-field ::=
    sorting-IRD-schema-descriptor
    | simple-attribute-type

IRD-view-name ::=
    assigned-access-name

IRD-view-name-list ::=
    IRD-view-name
    [ , IRD-view-name-list ]

IRD-view-option ::=
    all-word
    | IRD-view-name-list

irds-command ::=
    IRD-schema-command
    | IRD-command
    | general-command

irds-function ::=
    length-function
    | count-function
    | lines-function

irds-function-list ::=
    function-result
    [ , irds-function-list ]

irds-function-restriction-clause ::=
    irds-function
    relational-operator
    unsigned-integer

irds-module-1-reserved-word :=
    access-name-word
    all-word
    alternate-name-word
    and-word
    attributes-word
    descriptive-name-word
    entities-word
    entity-type-word
    exclude-word
    first-word
    for-word
    increment-word
    life-cycle-phase-word
    lines-word
    meta-entities-word
    meta-entity-type-word
    meta-relationships-word
    new-word
    not-word
    of-word
    only-word
    or-word
    order-word
    quality-word
    relationship-type-word
    relationships-word

```

APPENDIX

```

resequenced-word
revision-word
start-word
select-word
to-word
variation-word
version-word
with-word
xor-word

irds-name ::=
  spacing-character
  | letter-or-digit
  | irds-name-special-character
  { spacing-character | letter-or-digit |
    irds-name-special-character }

irds-name-special-character ::=
  & | % | # | $ | @

irds-word ::=
  access-name-word ::= ACCESS-NAME
  activate-word ::= ACTIVATE
  add-word ::= ADD
  after-word ::= AFTER
  all-word ::= ALL
  alternate-name-word ::=ALTERNATE-NAME
  and-word ::= AND
  ascending-word ::= ASCENDING
  ASN-word ::= ASN
  assigned-word ::= ASSIGNED
  attribute-group-type-word ::=
    ATTRIBUTE-GROUP-TYPE
  attributes-word ::= ATTRIBUTES
  attribute-type-word ::= ATTRIBUTE-TYPE
  attribute-type-validation-data-word ::=
    ATTRIBUTE-TYPE-VALIDATION-DATA
  attribute-type-validation-procedure-word ::=
    ATTRIBUTE-TYPE-VALIDATION-PROCEDURE
  check-word ::= CHECK
  command-imperative-word ::= COMMAND-IMPERATIVE
  context-word ::= CONTEXT
  controlled-word ::= CONTROLLED
  copy-word ::= COPY
  create-word ::= CREATE
  cumulative-word ::= CUMULATIVE
  current-date-word ::= CURRENT-DATE
  current-time-word ::= CURRENT-TIME
  current-word ::= CURRENT
  deactivate-word ::= DEACTIVATE
  decoded-word ::= DECODED
  defaults-word ::= DEFAULTS
  delete-word ::= DELETE
  descending-word ::= DESCENDING
  descriptive-name-word ::= DESCRIPTIVE-NAME
  directly-word ::= DIRECTLY
  each-word ::= EACH
  encoded-word ::= ENCODED
  entities-word ::= ENTITIES
  entity-word ::= ENTITY
  entity-type-word ::= ENTITY-TYPE
  except-word ::= EXCEPT
  exclude-word ::= EXCLUDE
  exists-word ::= EXISTS
  exit-word ::= EXIT
  export-word ::= EXPORT
  file-word ::= FILE
  first-word ::= FIRST
  for-word ::= FOR
  format-word ::= FORMAT
  forward-word ::= FORWARD
  from-word ::= FROM
  help-word ::= HELP
  highest-word ::= HIGHEST
  impact-word ::= IMPACT
  import-word ::= IMPORT
  in-word ::= IN
  increment-word ::= INCREMENT
  indirectly-word ::= INDIRECTLY
  individual-word ::= INDIVIDUAL
  inverse-word ::= INVERSE
  IRD-partition-word ::= IRD-PARTITION
  IRD-schema-views-word ::= IRD-SCHEMA-VIEWS
  IRD-schema-word ::= IRD-schema
  IRD-view-word ::= IRD-VIEW
  IRD-word ::= IRD
  IRDS-defaults-word ::= IRDS-DEFAULTS
  IRDS-limits-word ::= IRDS-LIMITS
  IRDS-reserved-names-word ::=
    IRDS-RESERVED-NAMES
  is-word ::= IS
  last-word ::= LAST
  life-cycle-phase-word ::= LIFE-CYCLE-PHASE
  lines-word ::= LINES
  load-word ::= LOAD
  lowest-word ::= LOWEST
  maximum-word ::= MAXIMUM
  message-word ::= MESSAGE
  meta-attributes-word ::= META-ATTRIBUTES
  meta-entities-word ::= META-ENTITIES
  meta-entity-word ::= META-ENTITY
  meta-entity-type-word ::= META-ENTITY-TYPE
  meta-relationship-word ::= META-RELATIONSHIP
  meta-relationships-word ::= META-RELATIONSHIPS
  minimal-word ::= MINIMAL
  modify-word ::= MODIFY
  name-word ::= NAME
  names-word ::= NAMES

```



```

new-word ::= NEW
no-word ::= NO
not-word ::= NOT
of-word ::= OF
on-word ::= ON
only-word ::= ONLY
or-word ::= OR
order-word ::= ORDER
other-word ::= OTHER
output-word ::= OUTPUT
page-word ::= PAGE
panel-word ::= PANEL
profiles-word ::= PROFILES
quality-indicator-word ::= QUALITY-INDICATOR
quality-word ::= QUALITY
related-word ::= RELATED
relationship-class-type-word ::=
    RELATIONSHIP-CLASS-TYPE
relationship-word ::= RELATIONSHIP
relationship-type-word ::= RELATIONSHIP-TYPE
relationships-word ::= RELATIONSHIPS
replace-word ::= REPLACE
resequenced-word ::= RESEQUENCED
revision-word ::= REVISION
route-word ::= ROUTE
save-word ::= SAVE
select-word ::= SELECT
sequence-word ::= SEQUENCE
set-word ::= SET
show-word ::= SHOW
sort-word ::= SORT
source-word ::= SOURCE
start-word ::= START
status-word ::= STATUS
suffix-word ::= SUFFIX
syntax-word ::= SYNTAX
target-word ::= TARGET
through-word ::= THROUGH
to-word ::= TO
unlock-word ::= UNLOCK
using-word ::= USING
variation-names-data-word ::=
    VARIATION-NAMES-DATA
variation-word ::= VARIATION
version-word ::= VERSION
view-word ::= VIEW
views-word ::= VIEWS
where-word ::= WHERE
with-word ::= WITH
xor-word ::= XOR

is-in-class-specification ::=
    in-word
    relationship-class-type-or-
    inverse

is-type-or-in-class-specification ::=
    is-type-specification
    | is-in-class-specification

is-type-specification ::=
    =
    relationship-type-or-inverse

key-meta-attribute ::=
    simple-meta-attribute-clause

last-attribute-of-group ::=
    non-text-attribute

last-meta-attribute ::=
    meta-attribute

length-function ::=
    length-function-qualifier'LENGTH

length-function-qualifier ::=
    non-repeating-attribute-type-designator
    | non-repeating-meta-attribute-type
    | entity-access-name-word
    | meta-entity-access-name-word
    | entity-descriptive-name-word
    | meta-entity-descriptive-name-word

less-than-or-equal ::= <=

less-than ::= <

letter ::=
    upper-case-letter
    | lower-case-letter

letter-or-digit ::=
    letter
    | digit

line-group ::=
    string-literal
    [ , line-group ]

line-number-1 ::=
    unsigned-integer
    | first-word
    | last-word

```

## APPENDIX

```

line-number-2 ::=
    unsigned-integer
    | first-word
    | last-word
    | maximum-word

line-number-defaults ::=
    line-number-defaults-positional-format
    | line-number-defaults-non-positional-format

line-number-defaults-non-positional-format ::=
    ( first-default-clause
      [,] [second-default-clause] )

line-number-defaults-positional-format ::=
    ( start [, increment] )

line-number-increment-clause ::=
    increment-word
    =
    increment-value

line-qualification ::=
    line-range-clause
    | line-number-defaults

line-range-clause ::=
    ( line-number-1
      [ through-word
        line-number-2 ] )

lines-function ::=
    lines-function-qualifier' LINES

lines-function-qualifier ::=
    text-attribute-type-designator
    | text-meta-attribute-type

literal ::=
    numeric-literal | long-string-literal
    | short-string-literal | irds-name

load-IRD-clause ::=
    load-word
    [ IRD-word ]
    [ from-word ]
    file-word
    file-name
    [ location-clause ]

location-clause ::=
    /* implementor-defined-format */

long-string-literal ::=
    string-literal

lower-case-letter ::=
    a | b | c | d | e | f | g | h | i | j | k |
    l | m | n | o | p | q | r | s | t | u | v |
    w | x | y | z

message-identifier ::=
    /* implementor-defined */

message-option ::=
    message-word
    [ message-identifier ]

meta-attribute ::=
    short-string-literal
    | numeric-literal
    | irds-name
    | null-mark

meta-attribute-1 ::=
    meta-attribute

meta-attribute-group ::=
    ( meta-attribute-1 | null
      {, meta-attribute-i | null }
      [, last-meta-attribute ] )

meta-attribute-group-list ::=
    meta-attribute-group
    [ [,] meta-attribute-group-list ]

meta-attribute-group-restriction-clause ::=
    for-word
    meta-attribute-group-type
    component-restriction-expression

meta-attribute-group-type ::=
    repeating-meta-attribute-group-type
    | audit-meta-attribute-group-type

meta-attribute-i ::= meta-attribute

meta-attribute-occurrence-qualification ::=
    ( meta-attribute )

meta-attribute-or-meta-attribute-group ::=
    simple-meta-attribute-clause
    | modified-repeating-meta-attribute-clause
    | modified-meta-attribute-group-clause
    | modified-text-meta-attribute-clause
    | text-meta-attribute-resequence-clause

```

```

meta-attribute-or-meta-attribute-group-type ::=
    non-repeating-meta-attribute-type
    | repeating-meta-attribute-type
    | meta-attribute-group-type
    | text-meta-attribute-type-option

meta-attribute-restriction-clause ::=
    meta-attribute-type
    relational-operator
    meta-attribute

meta-attribute-type ::=
    non-repeating-meta-attribute-type
    | repeating-meta-attribute-type
    | component-meta-attribute-type
    | text-meta-attribute-type

meta-attribute-type-list ::=
    meta-attribute-or-meta-attribute-group-type
    [ , meta-attribute-type-list ]

meta-entity-1-access-name ::=
    meta-entity-access-name

meta-entity-2-access-name ::=
    meta-entity-access-name

meta-entity-access-name ::=
    new-meta-entity-access-name
    | existing-meta-entity-access-name

meta-entity-assigned-access-name ::=
    assigned-access-name
    inverse-name

meta-entity-descriptive-name ::=
    descriptive-name

meta-entity-descriptive-name-clause ::=
    meta-entity-word
    descriptive-name-word
    =
    assigned-descriptive-name

meta-entity-designator ::=
    meta-entity-assigned-access-name
    | meta-entity-substitute-name

meta-entity-name-list ::=
    meta-entity-access-name
    [ , meta-entity-name-list ]

meta-entity-selection-clause ::=
    select-word
    meta-entity-selection-option
    [ where-word
      restriction-expression ]

meta-entity-selection-option
    all-word
    | meta-entity-name-list

meta-entity-substitute-name ::=
    irds-name

meta-entity-type ::=
    entity-type-word
    | relationship-class-type-word
    | relationship-type-word
    | attribute-type-word
    | attribute-group-type-word
    | IRD-partition-word
    | quality-indicator-word
    | attribute-type-validation-procedure-word
    | attribute-type-validation-data-word
    | variation-names-data-word
    | IRDS-limits-word
    | IRDS-defaults-word
    | IRDS-reserved-names-word
    | names-word

meta-entity-type-clause ::=
    meta-entity-type-word
    =
    meta-entity-type

meta-entity-type-list ::=
    meta-entity-type
    [ , meta-entity-type-list ]

meta-entity-type-qualification-clause ::=
    for-word
    [ meta-entities-word
      of-word ]
    meta-entity-type-list

meta-entity-type-restriction-clause ::=
    meta-entity-type-word
    = meta-entity-type-list

meta-relationship-class-type ::=
    CONNECTS
    | CONTAINS
    | USES
    | MEMBER-OF

```

## APPENDIX

```

meta-relationship-identifier ::=
    meta-entity-1-access-name
    meta-relationship-type-or-class-type
    meta-entity-2-access-name
    [ key-meta-attribute ]

meta-relationship-meta-attributes-option ::=
    and-word
    all-or-no-word
    meta-attribute-word

meta-relationship-type ::=
    RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-
        CLASS-TYPE
    | RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE
    | RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-
        TYPE
    | RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE
    | ENTITY-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE
    | ENTITY-TYPE-CONTAINS-ATTRIBUTE-TYPE
    | ENTITY-TYPE-USES-VARIATION-NAMES-DATA
    | ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE
    | ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-VALIDATION-
        DATA
    | ATTRIBUTE-TYPE-USES-ATTRIBUTE-TYPE-
        VALIDATION-PROCEDURE

meta-relationship-type-or-class-type ::=
    meta-relationship-type
    | meta-relationship-class-type

meta-relationship-type-qualification-clause ::=
    for-word
    meta-relationships-word
    meta-relationship-type

meta-relationships-existence-restriction-clause ::=
    [ no-word ]
    meta-relationships-word
    meta-relationship-type
    exists-word

minute-code ::=
    00|01|02|03|04|05|06|07|08|09|
    10|11|12|13|14|15|16|17|18|19|
    20|21|22|23|24|25|26|27|28|29|
    30|31|32|33|34|35|36|37|38|39|
    40|41|42|43|44|45|46|47|48|49|
    50|51|52|53|54|55|56|57|58|59

modified-attribute-or-attribute-group ::=
    simple-attribute-clause
    | modified-repeating-attribute-clause
    | modified-simple-attribute-group-clause
    | modified-repeating-attribute-group-clause
    | modified-text-attribute-clause
    | text-attribute-resequence-clause

modified-attribute-or-attribute-group-list ::=
    modified-attribute-or-attribute-group
    [ , modified-attribute-or-attribute-
        group-list ]

modified-attributes ::=
    modified-attribute-or-attribute-group-list

modified-attributes-and-or-order-clause ::=
    new-sequence-attribute-or-order-clause
    | modified-attributes
    | new-sequence-attribute-or-order-clause
    modified-attributes

modified-entity-attributes-clause ::=
    with-word
    [ attributes-word ]
    modified-attribute-or-attribute-group-list

modified-meta-attribute-clause-list ::=
    meta-attribute-or-meta-attribute-group
    [ modified-meta-attribute-clause-list ]

modified-meta-attribute-group-clause ::=
    [ all-word-or-new-word ]
    meta-attribute-group-type
    [ existing-group-occurrence ]
    =
    meta-attribute-group-list | null-mark

modified-meta-attributes-clause ::=
    with-word
    [ meta-attributes-word ]
    modified-meta-attribute-clause-list

modified-relationship-attributes-clause ::=
    with-word
    [ attributes-word ]
    modified-attributes-and-or-order-clause

```

```

modified-repeating-attribute-clause ::=
    [ all-word-or-new-word ]
    attribute-type-designator
    [ attribute-occurrence-qualification ]
    =
    repeating-attribute-list
    | null-mark

modified-repeating-attribute-group-clause ::=
    [ all-word-or-new-word ]
    attribute-group-type-designator
    [ group-occurrence-qualification ]
    =
    attribute-group-list

modified-repeating-meta-attribute-clause ::=
    [ all-word-or-new-word ]
    repeating-meta-attribute-type
    [ meta-attribute-occurrence-qualification ]
    =
    repeating-meta-attribute-list

modified-simple-attribute-group-clause ::=
    [ all-word ]
    attribute-group-type-designator
    =
    attribute-group
    | null-mark

modified-text-attribute-clause ::=
    [ all-word ]
    attribute-type-designator
    [ line-qualification ]
    =
    line-group | null-mark
    { subsequent-lines-subclause }

modified-text-meta-attribute-clause ::=
    [ all-word ]
    text-meta-attribute-type
    [ line-qualification ]
    =
    line-group | null-mark
    { subsequent-lines-subclause }

modify-entity-access-name-command ::=
    modify-word
    entity-word
    access-name-word
    from-word
    existing-entity-assigned-access-name
    to-word
    new-entity-assigned-access-name
    ;

modify-entity-command ::=
    modify-word
    entity-word
    existing-entity-access-name
    [ entity-type-clause ]
    [ new-version-clause ]
    [ entity-descriptive-name-declaration-clause ]
    [ quality-indicator-designation-clause ]
    [ modified-entity-attributes-clause ]
    ;

modify-entity-descriptive-name-command ::=
    modify-word
    entity-word
    descriptive-name-word
    existing-entity-specification
    to-word
    new-entity-assigned-descriptive-name
    ;

modify-entity-life-cycle-phase-command ::=
    modify-word
    entity-word
    life-cycle-phase-word
    [ for-word ]
    entity-name-list
    from-word
    current-life-cycle-phase
    to-word
    new-life-cycle-phase
    ;

modify-meta-entity-command ::=
    modify-word
    meta-entity-word
    existing-meta-entity-access-name
    [ meta-entity-type-clause ]
    [ new-meta-entity-version-clause ]
    [ meta-entity-descriptive-name-clause ]
    modified-meta-attributes-clause
    ;

```

APPENDIX

```

modify-meta-entity-life-cycle-phase-command ::=
  modify-word
  meta-entity-word
  life-cycle-phase-word
  [ for-word ]
  meta-entity-name-list
  from-word
  current-IRD-schema-life-cycle-phase
  to-word
  new-IRD-schema-life-cycle-phase
  [ implementation-defined-options ]
  ;

modify-meta-entity-access-name-command ::=
  modify-word
  meta-entity-word
  access-name-word
  from-word
  existing-meta-entity-assigned-access-name
  to-word
  new-meta-entity-assigned-access-name
  ;

modify-meta-entity-descriptive-name-command ::=
  modify-word
  meta-entity-word
  descriptive-name-word
  existing-meta-entity-specification
  to-word
  new-meta-entity-assigned-descriptive-name
  ;

modify-meta-relationship-command ::=
  modify-word
  meta-relationship-identifier
  modified-meta-attributes-clause
  ;

modify-relationship-command ::=
  modify-word
  relationship-word
  relationship-identification-clause
  [ modified-relationship-attributes-clause ]
  ;

month-code ::=
  01|02|03|04|05|06|07|08|09|10|11|12

new-entity ::=
  new-entity-access-name
  | null-mark
  | new-version-clause

new-entity-access-name ::= access-name

new-entity-assigned-access-name ::=
  assigned-access-name

new-entity-assigned-descriptive-name ::=
  assigned-descriptive-name

new-entity-attributes-clause ::=
  with-word
  [ attributes-word ]
  new-attribute-or-attribute-group-list

new-entity-name ::=
  new-entity-access-name
  | null-mark

new-IRD-schema-life-cycle-phase ::=
  IRD-schema-life-cycle-phase

new-life-cycle-phase ::= irds-name

new-line ::=
  /* implementation-dependent end-of-line
  indicator */

new-meta-attribute-clause-list ::=
  new-meta-attribute-or-meta-attribute-group
  [ new-meta-attribute-clause-list ]

new-meta-attribute-group-clause ::=
  meta-attribute-group-type
  =
  meta-attribute-group-list

new-meta-attribute-or-meta-attribute-group ::=
  simple-meta-attribute-clause
  | new-repeating-meta-attribute-clause
  | new-meta-attribute-group-clause
  | new-text-meta-attribute-clause

new-meta-attributes-clause ::=
  with-word
  [ meta-attributes-word ]
  new-meta-attribute-clause-list

new-meta-entity-access-name ::= access-name

new-meta-entity-assigned-access-name ::=
  assigned-access-name

new-meta-entity-assigned-descriptive-name ::=
  assigned-descriptive-name

```

```

new-meta-entity-specification ::=
    new-meta-entity-access-name
    | new-meta-entity-version-clause

new-meta-entity-version-clause ::=
    new-word
    meta-entity-word
    version-word
    [ = version-identifier ]

new-relationship-attributes-clause ::=
    with-word
    [attributes-word]
    new-attribute-or-attribute-group-list

new-relationship-between-existing-entities ::=
    entity-1-name
    relationship-type-or-relationship-class-type
    entity-2-name
    [ order-clause-or-sequence-attribute ]

new-relationship-identification-clause ::=
    new-relationship-between-existing-entities
    | relate-existing-entity-to-new-entity

new-repeating-attribute-clause ::=
    attribute-type-designator
    =
    repeating-attribute-list

new-repeating-attribute-group-clause ::=
    attribute-group-type-designator
    =
    attribute-group-list

new-repeating-meta-attribute-clause ::=
    repeating-meta-attribute-type
    = repeating-meta-attribute-list

new-sequence-attribute-or-order-clause ::=
    sequence-attribute
    | order-clause

new-simple-attribute-group-clause ::=
    attribute-group-type-designator
    = attribute-group

new-text-attribute-clause ::=
    attribute-type-designator
    [line-number-defaults]
    =
    string-literal-list

new-text-meta-attribute-clause ::=
    text-meta-attribute-type
    [ line-number-defaults ]
    =
    string-literal-list

new-version-clause ::=
    new-word
    entity-word
    version-word
    [ = version-identifier ]

no-attributes-phrase ::=
    no-word
    attributes-word

non-fractional-real ::=
    unsigned-integer
    exponent-character
    scale-factor

non-repeating-attribute-type-designator ::=
    attribute-type-designator

non-repeating-meta-attribute-type ::=
    simple-meta-attribute-type
    | system-maintained-meta-attribute-type

non-text-attribute ::=
    short-string-literal
    | irds-name
    | numeric-literal
    | date-time-literal
    | null-mark

not-equal ::= <>

null-mark ::=
    /* implementation-selected-special-character
    */

null-string ::=
    /* A quote-character immediately followed by
    the same quote-character */

number-scan-mask ::=
    numeric-literal | *

numeric-literal ::=
    integer-literal
    | real-literal

```

## APPENDIX

```

only-entity-types-option ::=
    only-word
    entity-type-list

only-specified-attributes-phrase ::=
    only-word
    specified-attributes-phrase

option ::=
    effective-view-option
    | attribute-decoding-option
    | implementor-defined-option

options-list ::=
    option
    [ options-list ]

order-clause ::=
    order-word
    =
    unsigned-integer

order-clause-or-sequence-attribute ::=
    order-clause
    | sequence-attribute

order-restriction-clause ::=
    order-word
    relational-operator
    unsigned-integer

other-IRD-schema-clause ::=
    [ source-or-target-option ]
    IRD-schema-word
    [ is-word ]
    other-IRD-schema-location

other-IRD-schema-location ::=
    in-IRD-clause
    | in-file-clause
    | minimal-word

output-impact-of-change-command ::=
    output-word
    [ impact-option ]
    impact-word
    [ using-IRD-views-clause ]
    selection-option
    [ sort-clause ]
    [ impact-show-options ]
    [ route-clause ]
    [ implementor-defined-clauses ]
    ;

output-syntax-command ::=
    output-word
    syntax-word
    [ using-IRD-views-clause ]
    selection-option
    [ sort-clause ]
    [ syntax-show-options ]
    [ route-clause ]
    [ implementor-defined-clauses ]
    ;

panel-name ::=
    /* implementor-defined-format */

panel-name-option ::=
    name-word
    =
    panel-name

panel-transfer-option ::=
    panel-name-option
    | command-imperative

path ::=
    forward-association-designator
    | inverse-association-designator

path-list ::=
    path
    [ , path-list ]

qualification-clause ::=
    entity-type-qualification-clause
    | relationship-type-qualification-clause

qualified-impacted-entity-show-clause ::=
    entity-type-qualification-clause
    impacted-entity-show-clause-list

qualified-show-clause ::=
    entity-type-qualification-clause
    show-options-within-entity-type

qualified-show-clause-list ::=
    qualified-show-clause
    [ qualified-show-clause-list ]

quality-indicator-designation-clause ::=
    quality-word
    =
    quality-indicator-designator

quality-indicator-designator ::=
    meta-entity-designator

```



```

quality-indicator-restriction-clause ::=
    quality-word
    equal-or-not-equal
    quality-indicator-designator

quote-character ::= ' | "

real-literal ::=
    signed-real
    | unsigned-real

regular-token ::=
    irds-name
    | scan-mask
    | irds-word
    | numeric-literal

relate-existing-entity-to-new-entity ::=
    entity-1-name
    relationship-type-or-relationship-class-type
    new-word
    [ entity-2-entity-type-designator ]
    entity-2-name-or-null-mark
    [ order-clause-or-sequence-attribute ]

related-entities-clause ::=
    entities-word
    [ directly-word ]
    related-word
    to-word
    entity-access-names-or-list
    [ via-word
      path-list ]

relational-operator ::=
    equals
    | greater-than
    | less-than
    | greater-than-or-equal
    | less-than-or-equal
    | not-equal

relationship-class-type-designator ::=
    meta-entity-designator

relationship-class-type-inverse-name ::=
    inverse-name

relationship-class-type-or-inverse ::=
    relationship-class-type-designator
    relationship-class-type-inverse-name

relationship-display-option ::=
    all-relationships-option
    | excluding-relationships-option
    | specified-relationship-option

relationship-existence-restriction-clause ::=
    [ no-word ]
    [ forward-inverse-criterion ]
    relationships-word
    [ of-word
    [ relationship-type-word ]
    relationship-type-or-inverse ]
    exists-word

relationship-identification-clause ::=
    entity-1-name
    relationship-type-or-relationship-class-type
    entity-2-name
    [ order-clause-or-sequence-attribute ]

relationship-identification-list ::=
    relationship-identification-clause
    [ [, ] relationship-identification-list ]

relationship-restriction-clause ::=
    relationship-type-restriction-clause
    | attribute-restriction-clause
    | attribute-group-restriction-clause
    | order-restriction-clause
    | text-attribute-substring-restriction-clause
    | irds-function-restriction-clause

relationship-restriction-expression ::=
    relationship-restriction-clause
    | ( relationship-restriction-expression )
    | relationship-restriction-expression
    boolean-operator
    relationship-restriction-expression

relationship-selection-clause ::=
    select-word
    [ all-word
    [ forward-or-inverse-word ]
    ]
    relationships-word
    for-word
    specified-entities
    [ where-word
      relationship-restriction-expression ]

relationship-syntax-option ::=
    for-each-phrase
    | after-last-phrase

```

## APPENDIX

```

relationship-type-designator ::=
    meta-entity-designator

relationship-type-inverse-name ::=
    inverse-name

relationship-type-list ::=
    relationship-type-designator
    [ , relationship-type-list ]

relationship-type-or-class-type ::=
    relationship-type-designator
    | relationship-class-type-designator

relationship-type-or-class-type-list ::=
    relationship-type-or-class-type
    [ , relationship-type-or-class-type-list ]

relationship-type-or-inverse ::=
    relationship-type-designator
    relationship-type-inverse-name

relationship-type-or-relationship-class-type ::=
    relationship-type-designator
    | relationship-class-type-designator

relationship-type-qualification-clause ::=
    for-word
    [ relationships-word
      of-word ]
    relationship-type-list

relationship-type-qualified-shows ::=
    relationship-type-qualification-clause
    show-clause-options

relationship-type-restriction-clause ::=
    relationship-type-word
    is-type-or-in-class-specification

relationship-type-selection ::=
    relationship-type-designator
    | relationship-type-inverse-name
    | relationship-class-type-designator
    | relationship-class-type-inverse-name

relationship-type-selection-list ::=
    relationship-type-selection
    [ , relationship-type-selection-list ]

relationships-identification ::=
    relationship-selection-clause
    | relationship-identification-list

repeating-attribute-list ::=
    non-text-attribute
    [ , repeating-attribute-list ]

repeating-meta-attribute-group-type ::=
    DATA-RANGE
    | DATA-VALUE

repeating-meta-attribute-list ::=
    meta-attribute
    [ , repeating-meta-attribute-list ]

repeating-meta-attribute-type ::=
    INVERSE-NAME
    | META-ENTITY-SUBSTITUTE-NAME
    | PICTURE
    | RESERVED-ENTITY-NAME
    | RESERVED-META-ENTITY-NAME
    | VARIATION

restore-IRD-schema-command ::=
    restore-word
    IRD-schema-word
    ;

restriction-clause ::=
    entity-type-restriction-clause
    | relationship-existence-restriction-clause
    | attribute-restriction-clause
    | attribute-group-restriction-clause
    | text-attribute-substring-restriction-clause
    | order-restriction-clause
    | IRD-life-cycle-phase-restriction-clause
    | audit-attribute-restriction-clause
    | alternate-name-restriction-clause
    | quality-indicator-restriction-clause
    | irds-function-restriction-clause
    | entity-assigned-access-name-restriction-clause
    | entity-assigned-descriptive-name-
      restriction-clause
    | variation-name-restriction-clause
    | revision-number-restriction-clause

restriction-expression ::=
    restriction-expression
    boolean-operator
    restriction-expression
    ( restriction-expression )
    [ qualification-clause ]
    restriction-expression
    | restriction-clause

revision-number ::= unsigned-integer

```

```

revision-number-restriction-clause ::=
    revision-word
    relational-operator
    comparison-value

route-clause ::=
    route-word
    to-word
    destination-list

scale-factor ::= signed-integer

scan-mask ::=
    name-scan-mask
    | number-scan-mask
    | string-scan-mask

Second-code ::=
    00|01|02|03|04|05|06|07|08|09|
    10|11|12|13|14|15|16|17|18|19|
    20|21|22|23|24|25|26|27|28|29|
    30|31|32|33|34|35|36|37|38|39|
    40|41|42|43|44|45|46|47|48|49|
    50|51|52|53|54|55|56|57|58|59

second-default-clause ::=
    start-line-number-clause
    | line-number-increment-clause

security-entity-name ::=
    assigned-access-name

selected-attribute-or-attribute-group ::=
    simple-attribute-type
    | attribute-group-phrase
    | text-attribute-phrase

selection-option ::=
    entity-selection-criteria-clause

sequence-attribute ::=
    simple-attribute-clause

session-status-command ::=
    status-word
    all-or-status-options

set-session-defaults-command ::=
    set-word
    options-list
    [ save-word ]
    ;

short-string-literal ::= string-literal

show-all-clause ::=
    show-word
    all-word

show-attributes-clause ::=
    show-word
    attribute-selection-option

show-clause-list ::=
    [ show-title-clause ]
    unqualified-show-clause-list
    [ qualified-show-clause-list ]

show-clause-options ::=
    show-clause-selection
    [ show-clause-options ]

show-clause-selection ::=
    show-IRD-life-cycle-phase-clause
    | show-entity-access-name-clause
    | show-entity-descriptive-name-clause
    | show-entity-type-clause
    | show-relationships-clause
    | show-attributes-clause
    | show-quality-indicator-clause

show-counts-clause ::=
    show-word
    irds-function-list

show-entity-access-name-clause ::=
    show-word
    entity-word
    access-name-display-option

show-entity-descriptive-name-clause ::=
    show-word
    entity-word
    descriptive-name-display-option

show-entity-type-clause ::=
    show-word
    entity-type-word

show-IRD-life-cycle-phase-clause ::=
    show-word
    IRD-word
    life-cycle-phase-word

show-IRD-schema-life-cycle-phase-clause ::=
    show-word
    IRD-schema-word
    life-cycle-phase-word

```

## APPENDIX

```

show-IRD-schema-options ::=
    [ show-title-clause ]
    IRD-schema-show-clause-list

show-meta-attributes-clause ::=
    show-word
    all-or-specified-option

show-meta-relationships-clause ::=
    show-word
    meta-relationships-word
    [ meta-relationship-meta-attributes-option ]

show-options ::=
    | show-predefined-display-clause
    | show-clause-list

show-options-within-entity-type ::=
    show-clause-options
    | relationship-type-qualified-shows
    | entity-and-relationship-type-qualified-shows

show-predefined-display-clause ::=
    show-word
    display-format

show-quality-indicator-clause ::=
    show-word
    quality-word

show-related-meta-entities-clause ::=
    show-word
    [ directly-or-indirectly-option ]
    related-word
    meta-entities-word
    [ where-word
        meta-entity-type-restriction-clause ]

show-relationship-syntax-clause ::=
    show-word
    relationship-word
    syntax-word
    relationship-syntax-option

show-relationships-clause ::=
    show-word
    relationship-display-option
    [ attribute-display-option ]

show-IRD-schema-life-cycle-phase-clause ::=
    show-word
    IRD-schema-word
    life-cycle-phase-word

show-title-clause ::=
    show-word
    short-string-literal
    [ first-or-each-page-option ]

signed-integer ::=
    [ + | - ] unsigned-integer

signed-real ::=
    [ + | - ] unsigned-real

simple-attribute-clause ::=
    attribute-type-designator
    =
    non-text-attribute

simple-attribute-type ::=
    attribute-type-designator

simple-meta-attribute-clause ::=
    simple-meta-attribute-type
    =
    meta-attribute

simple-meta-attribute-type ::=
    ENTITY-CLASS
    | FORMAT
    | GROUP-POSITION
    | INTEGER-LIMIT
    | LINE-COUNT-LIMIT
    | LINE-LENGTH-LIMIT
    | MAXIMUM-ATTRIBUTE-LENGTH
    | MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH
    | MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-
        DEFAULT
    | MAXIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-
        LIMIT
    | MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-
        LENGTH
    | MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-
        LENGTH-DEFAULT
    | MAXIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-
        LENGTH-LIMIT
    | MAXIMUM-META-ENTITY-ASSIGNED-ACCESS-NAME-
        LENGTH-LIMIT
    | MAXIMUM-META-ENTITY-ASSIGNED-DESCRIPTIVE-
        NAME-LENGTH-LIMIT
    | MAXIMUM-NUMBER-OF-OCCURRENCES
    | MAXIMUM-NUMBER-OF-OCCURRENCES-DEFAULT
    | MAXIMUM-NUMBER-OF-OCCURRENCES-LIMIT
    | MINIMUM-ATTRIBUTE-LENGTH
    | MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH
    | MINIMUM-ENTITY-ASSIGNED-ACCESS-NAME-LENGTH-
        DEFAULT

```

```

| MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-
      LENGTH
| MINIMUM-ENTITY-ASSIGNED-DESCRIPTIVE-NAME-
      LENGTH-DEFAULT
| ORIGIN
| STANDARD-MODE
| POSITION
| SEQUENCE-PARAMETER
| SEQUENCED
| SIGNIFICANT-ATTRIBUTES
| SINGULAR
| START-NAME
| STRING-LENGTH-LIMIT
| SYSTEM-GENERATED
| VALIDATION-TYPE
| VARIATION-LENGTH-LIMIT

sort-clause ::=
  sort-word
  sequence-word
  =
  sort-parm-list

sort-meta-entities-clause ::=
  sort-word
  sequence-word
  =
  IRD-schema-sort-parameter-list

sort-parm ::=
  sort-parm-form-1
  | sort-parm-form-2

sort-parm-form-1 ::=
  IRD-sort-field

sort-parm-form-2 ::=
  ( IRD-sort-field
    , ascending-or-descending )

sort-parm-list ::=
  sort-parm
  [ , sort-parm-list ]

sorting-IRD-schema-descriptor ::=
  life-cycle-phase-word
  | entity-type-word
  | version-word
  | variation-word
  | revision-word
  | entity-access-name-phrase
  | entity-descriptive-name-phrase

source-or-target-option ::=
  source-word
  | target-word

space ::= /* blank character */

spacing-character ::= - | _

special-character ::=
  + | - | * | / | = | $ | , | . | : | ; | ( | )
  | & | ^ | % | # | @ | ! | > | < | ? | _ | { | }
  | [ | ] | \ | ~ | vertical-bar | new-line
  | space | quote-character

specified-attributes-phrase ::=
  attributes-word
  attribute-selection-list

specified-date ::=
  standard-format-date
  | /* implementor-defined-format */

specified-entities ::=
  name-scan-pattern-list

specified-meta-attributes-option ::=
  meta-attributes-word
  meta-attribute-type-list

specified-relationships-option ::=
  relationships-word
  relationship-type-selection-list

specified-time ::=
  standard-format-time
  | /* implementor-defined-format */

standard-display-format-option ::=
  syntax-format
  | individual-impact-format
  | cumulative-impact-format

standard-format-date ::=
  [ century-code ]
  year-code
  month-code
  day-code

standard-format-time ::=
  hour-code
  minute-code
  [second-code]

```

## APPENDIX

```

start ::=
    unsigned-integer
    | null

start-line-number-clause ::=
    start-word
    =
    starting-line-number

starting-line-number ::=
    unsigned-integer

static-meta-attribute-type ::=
    COMMON-TO-ENTITY-TYPES
    | CONNECTABLE
    | TEXT-IN-GROUPS-ALLOWED

status-option ::=
    IRD-word
    | views-option
    | defaults-word
    | implementor-defined-option

status-option-list ::=
    status-option
    [ , status-option-list ]

string-literal ::=
    quote-character {character}
    quote-character

string-literal-list ::=
    string-literal
    [ , string-literal-list ]

string-scan-mask ::=
    substitution-character
    | short-string-literal
    | irds-name-special-character
    { substitution-character
    | short-string-literal
    | irds-name-special-character }

subsequent-lines-subclause ::=
    lines-word
    =
    line-range-clause
    line-group | null-mark

substitution-character ::= * | ?

syntax-format ::=
    syntax-word
    syntax-show-options

syntax-show-options ::=
    [ show-title-clause ]
    [ show-IRD-life-cycle-phase-clause ]
    [ show-relationships-clause ]
    [ show-relationship-syntax-clause ]

system-maintained-meta-attribute-type ::=
    audit-meta-attribute-type
    | static-meta-attribute-type
    | control-meta-attribute-type

test-value ::=
    numeric-literal
    | short-string-literal
    | irds-name
    | null-mark
    | name-scan-mask
    | number-scan-mask
    | string-scan-mask

test-value-list ::=
    test-value
    [ , test-value-list ]

text-attribute ::=
    string-literal-list

text-attribute-phrase ::=
    text-attribute-type
    [ line-range-clause ]

text-attribute-resequence-clause ::=
    text-attribute-type-designator
    [ line-range-clause ]
    =
    resequenced-word
    [ start-line-number-clause ]
    [ line-number-increment-clause ]

text-attribute-substring-restriction-clause ::=
    attribute-type-designator
    [ line-range-clause ]
    equals-or-not-equal
    string-scan-mask

text-attribute-type ::=
    attribute-type-designator

text-attribute-type-designator ::=
    attribute-type-designator

```

```

text-meta-attribute-resequence-clause ::=
    text-meta-attribute-type
    [ line-range-clause ]
    =
    resequenced-word
    [ start-line-number-clause ]
    [ line-number-increment-clause ]

text-meta-attribute-type ::=
    DESCRIPTION-OF-RULES
    | PURPOSE

text-meta-attribute-type-option ::=
    text-meta-attribute-type
    [ line-range-clause ]

time-literal ::=
    specified-time
    | current-time-word

token ::=
    regular-token
    | delimiter-token

type-designator ::=
    entity-type-designator
    | relationship-type-designator
    | relationship-class-type-designator
    | attribute-type-designator
    | attribute-group-type-designator

unqualified-show-clause-list ::=
    general-output-show-clause-option
    [ unqualified-show-clause-list ]

unsigned-integer ::=
    digit { digit }

unsigned-real ::=
    non-fractional-real
    | fractional-real

upper-case-letter ::=
    A | B | C | D | E | F | G | H | I | J | K | L
    | M | N | O | P | Q | R | S | T | U | V | W | X
    | Y | Z

using-entity-assigned-access-name-specification ::=
    for-word
    | existing-entity-assigned-access-name

using-entity-assigned-descriptive-name-specification ::=
    from-word
    | existing-entity-assigned-descriptive-name

using-IRD-schema-views-clause ::=
    using-word
    IRD-schema-views-word
    IRD-schema-view-list

using-IRD-views-clause ::=
    using-word
    IRD-view-word
    =
    IRD-view-option

using-meta-entity-assigned-access-name-specification ::=
    for-word
    existing-meta-entity-assigned-access-name

using-meta-entity-assigned-descriptive-name-specification ::=
    from-word
    existing-meta-entity-assigned-descriptive-name

utility-command ::=
    set-session-defaults-command
    | session-status-command
    | help-command
    | exit-IRDS-system-command
    | enter-panel-dialogue-command

variation-name ::=
    letter
    [ irds-name ]
    | null

variation-name-restriction-clause ::=
    variation-word
    relational-operator
    irds-name

version-identifier ::=
    ( variation-name : revision-number )
    | ( variation-name )
    | ( revision-number )

vertical-bar ::= |

views-option ::=
    IRD-word | IRD-schema-word
    views-word

```

## APPENDIX

```
with-meta-relationships-clause ::=  
  with-word  
  meta-relationships-word
```

```
with-relationships-clause ::=  
  with-word  
  relationships-word
```

```
year-code ::= digit  
           digit
```



## Chapter 2

### Module 2 - Basic Functional IRD Schema

#### CONTENTS

SECTION	PAGE
1 Scope, Purpose, and Application . . . . .	2-1
1.1 Scope . . . . .	2-1
1.2 Purpose . . . . .	2-1
1.3 Conformance . . . . .	2-2
1.4 Organization . . . . .	2-2
1.5 Notation . . . . .	2-3
2 References . . . . .	2-3
3 Definitions . . . . .	2-3
4 Module Overview . . . . .	2-3
5 The Basic Functional IRD Schema . . . . .	2-4
5.1 Entity-Types in the Basic Functional IRD Schema . . . . .	2-4
5.1.1 Document . . . . .	2-5
5.1.2 Element . . . . .	2-5
5.1.3 File . . . . .	2-5
5.1.4 Module . . . . .	2-6
5.1.5 Program . . . . .	2-6
5.1.6 Record . . . . .	2-6
5.1.7 System . . . . .	2-7
5.1.8 User . . . . .	2-7
5.2 Relationship-Types in the Basic Functional IRD Schema . . . . .	2-7
5.2.1 Document Contains Document . . . . .	2-8
5.2.2 Document Contains Element . . . . .	2-8
5.2.3 Document Contains Record . . . . .	2-8
5.2.4 Document Derived From Document . . . . .	2-9
5.2.5 Document Derived From File . . . . .	2-9
5.2.6 Document Derived From Record . . . . .	2-9
5.2.7 Element Contains Element . . . . .	2-10
5.2.8 Element Derived From Document . . . . .	2-10
5.2.9 Element Derived From Element . . . . .	2-10
5.2.10 Element Derived From File . . . . .	2-10
5.2.11 Element Derived From Record . . . . .	2-11
5.2.12 Element Standard For Element . . . . .	2-11
5.2.13 File Contains Document . . . . .	2-11
5.2.14 File Contains Element . . . . .	2-12
5.2.15 File Contains File . . . . .	2-12
5.2.16 File Contains Record . . . . .	2-12
5.2.17 File Derived From Document . . . . .	2-13
5.2.18 File Derived From File . . . . .	2-13
5.2.19 File Has Access Key Element . . . . .	2-13

SECTION	PAGE	
5.2.20	File Has Sort Key Element . . . . .	2-13
5.2.21	Module Calls Module . . . . .	2-14
5.2.22	Module Contains Module . . . . .	2-14
5.2.23	Module Goes To Module . . . . .	2-14
5.2.24	Module Processes Document . . . . .	2-15
5.2.25	Module Processes Element . . . . .	2-15
5.2.26	Module Processes File . . . . .	2-15
5.2.27	Module Processes Record . . . . .	2-16
5.2.28	Program Calls Module . . . . .	2-16
5.2.29	Program Calls Program . . . . .	2-16
5.2.30	Program Contains Module . . . . .	2-16
5.2.31	Program Contains Program . . . . .	2-17
5.2.32	Program Goes To Program . . . . .	2-17
5.2.33	Program Processes Document . . . . .	2-17
5.2.34	Program Processes Element . . . . .	2-18
5.2.35	Program Processes File . . . . .	2-18
5.2.36	Program Processes Record . . . . .	2-18
5.2.37	Record Contains Element . . . . .	2-19
5.2.38	Record Contains Record . . . . .	2-19
5.2.39	Record Derived From Document . . . . .	2-19
5.2.40	Record Derived From File . . . . .	2-19
5.2.41	Record Derived From Record . . . . .	2-20
5.2.42	System Contains Module . . . . .	2-20
5.2.43	System Contains Program . . . . .	2-20
5.2.44	System Contains System . . . . .	2-21
5.2.45	System Goes To System . . . . .	2-21
5.2.46	System Processes Document . . . . .	2-21
5.2.47	System Processes Element . . . . .	2-22
5.2.48	System Processes File . . . . .	2-22
5.2.49	System Processes Record . . . . .	2-22
5.2.50	User Processes Document . . . . .	2-22
5.2.51	User Processes Element . . . . .	2-23
5.2.52	User Processes File . . . . .	2-23
5.2.53	User Processes Record . . . . .	2-23
5.2.54	User Responsible For Document . . . . .	2-24
5.2.55	User Responsible For Element . . . . .	2-24
5.2.56	User Responsible For File . . . . .	2-24
5.2.57	User Responsible For Module . . . . .	2-25
5.2.58	User Responsible For Program . . . . .	2-25
5.2.59	User Responsible For Record . . . . .	2-25
5.2.60	User Responsible For System . . . . .	2-25
5.2.61	User Runs Module . . . . .	2-26
5.2.62	User Runs Program . . . . .	2-26
5.2.63	User Runs System . . . . .	2-26
5.3	Relationship-Class-Types in the Basic Functional IRD Schema . . . . .	2-27
5.3.1	Calls . . . . .	2-27
5.3.2	Contains . . . . .	2-28
5.3.3	Derived From . . . . .	2-28
5.3.4	Goes To . . . . .	2-28
5.3.5	Processes . . . . .	2-29
5.3.6	Responsible For . . . . .	2-29

SECTION	PAGE
5.3.7	Runs . . . . . 2-29
5.4	Attribute-Types in the Basic Functional IRD Schema . . . . . 2-29
5.4.1	Access Method . . . . . 2-30
5.4.2	Allowable Value . . . . . 2-31
5.4.3	Alternate Name . . . . . 2-31
5.4.4	Alternate Name Context . . . . . 2-31
5.4.5	Classification . . . . . 2-31
5.4.6	Code List Location . . . . . 2-32
5.4.7	Comments . . . . . 2-32
5.4.8	Data Class . . . . . 2-32
5.4.9	Data Type . . . . . 2-33
5.4.10	Description . . . . . 2-33
5.4.11	Document Category . . . . . 2-33
5.4.12	Duration Type . . . . . 2-33
5.4.13	Duration Value . . . . . 2-34
5.4.14	External Security . . . . . 2-34
5.4.15	Frequency . . . . . 2-34
5.4.16	High Of Range . . . . . 2-35
5.4.17	Internal Format . . . . . 2-35
5.4.18	Justification . . . . . 2-35
5.4.19	Length . . . . . 2-36
5.4.20	Location . . . . . 2-36
5.4.21	Low Of Range . . . . . 2-36
5.4.22	Number Of Lines Of Code . . . . . 2-37
5.4.23	Number Of Records . . . . . 2-37
5.4.24	Precision . . . . . 2-37
5.4.25	Record Category . . . . . 2-37
5.4.26	Relative Position . . . . . 2-38
5.4.27	Scale . . . . . 2-38
5.4.28	System Category . . . . . 2-38
5.4.29	Usage . . . . . 2-39
5.5	Attribute-Group-Types in the Basic Functional IRD Schema . . . . . 2-39
5.5.1	Allowable Range . . . . . 2-40
5.5.2	Duration . . . . . 2-40
5.5.3	Identification Names . . . . . 2-40
6	Associations between Basic Functional IRD Schema Descriptors . . . . . 2-41
6.1	Entity-Types Associated with Relationship-Types . . . . . 2-41
6.2	Relationship-Types Associated with Relationship-Class-Types . . . . . 2-41
6.3	Attribute-Types Associated with Entity-Types . . . . . 2-46
6.4	Attribute-Group-Types Associated with Entity-Types . . . . . 2-47
6.5	Attribute-Types Associated with Relationship-Types . . . . . 2-47
6.6	Attribute-Group-Types Associated with Relationship-Types . . . . . 2-49
6.7	Attribute-Types Associated with Attribute-Group-Types . . . . . 2-49
Tables	
Table 1	- Association of Entity-Types and Relationship-Types (Page 1) . . . . . 2-42
Table 1	- Association of Entity-Types and Relationship-Types (Page 2) . . . . . 2-43
Table 1	- Association of Entity-Types and Relationship-Types (Page 3) . . . . . 2-44
Table 2	- Association of Entity-Types and Attribute-Types . . . . . 2-48
Table 3	- Association of Entity-Types and Attribute-Group-Types . . . . . 2-49



## Chapter 2

### Module 2 - Basic Functional IRD Schema

#### 1 Scope, Purpose, and Application

##### 1.1 Scope

Module 2 of the standard IRDS specifies those entity-types, relationship-types, relationship-class-types, attribute-types, and attribute-group-types which make up the Basic Functional IRD Schema.

The Basic Functional IRD Schema defined in this Module extends the Minimal IRD Schema specified in Section 10 of Module 1 of the standard IRDS.

##### 1.2 Purpose

The need to specify a Starter Set of IRD Schema descriptors has been a consistent requirement throughout the development of the standard IRDS. The need is based on the assumption that many IRDS installations would not have the necessary initial understanding and background to establish, immediately, a meaningful IRD Schema. This could result in false starts which could diminish the acceptability of the IRDS in the organization. This specification is intended to satisfy this "start-up" requirement. It is assumed that an installation will, as experience grows, use the IRD Schema maintenance commands to modify the IRD Schema, including the Basic Functional IRD Schema, to satisfy the installation's evolving requirements.

Assume that one wished to support installation requirements not supported by standard IRDS facilities and not supported by the installation's current IRD Schema. The installation would need to be able to modify the IRD Schema and would have to add new software facilities designed to operate against the modified IRD Schema. This is quite similar to adding a new application into a database environment. The principal difference is that, unlike a general database environment, the IRDS software assures the integrity of the existing and new IRD Schema, and the IRD, based on "rules" which exist in the Minimal IRD Schema.

To illustrate how one would "extend" the IRD Schema to support additional requirements, it is necessary to understand the character of the IRD Schema. Basically, the IRD Schema can be defined in terms of entities, relationships and attributes. However, because of the potential for misunderstanding which may occur when discussing the contents of the IRD Schema and the IRD, similar, yet distinct, terminology is used to describe the IRD and the IRD Schema. In particular,

- (1) The IRD is assumed to contain:
  - (a) Entities.
  - (b) Relationships between entities.
  - (c) Attributes and attribute-groups which document characteristics of the entities and the relationships.

- (2) Similarly, the IRD Schema is assumed to contain:
- (a) Entities, which we shall refer to as meta-entities.
  - (b) Relationships between meta-entities, which we shall refer to as meta-relationships.
  - (c) Attributes and attribute groups, which we shall refer to as meta-attributes and meta-attribute-groups, which document characteristics of the meta-entities and the meta-relationships.

The commands which operate against the IRD Schema are specified in Subsection 5.1 of Module 1. For example, a new entity-type or a new relationship-type can be added to the IRD Schema using the "add meta-entity" command, existing attribute-type-validation-data or variation-names-data can be modified using the "modify meta-entity" command, and attribute-types can be deleted from the IRD Schema using the "delete meta-entity" command. After the new entity-type or relationship-type is added to the IRD Schema, it is normally necessary and desirable to associate attribute-types with that new IRD Schema element. This is accomplished by using the "add meta-relationship" command; i.e., the fact that an attribute-type is associated with an entity- or relationship-type is represented by a meta-relationship in the IRD Schema. Of course, these commands shall not violate any integrity rules associated with existing IRD Schema descriptors or with the content of the IRD which is defined by those IRD Schema descriptors.

### 1.3 Conformance

The Definition of a Conformant Implementation of Module 2 of an American National Standard IRDS is given in the Definition of a Conformant Implementation of an American National Standard IRDS.

### 1.4 Organization

Module 2 of the standard IRDS consists of five major components:

- (1) References.
- (2) Definitions.
- (3) Module Overview.
- (4) The Basic Functional IRD Schema.
- (5) Associations between Basic Functional IRD Schema Descriptors.

References are provided in Section 2.

Definitions of terms for this Module are given in Section 3.

An overview for this Module is provided in Section 4.

A detailed specification for each new entity-type, relationship-type, relationship-class-type, attribute-type, and attribute-group-type in the Basic Functional IRD Schema is given in Section 5.

Associations exist between the IRD Schema descriptors of the Basic Functional IRD Schema. These associations are given in Section 6.

This Module should be reviewed sequentially.

## 1.5 Notation

Module 2 uses the same notational conventions as identified in Module 1 of the standard IRDS.

## 2 References

None

## 3 Definitions

There are no definitions which apply to this Module.

## 4 Module Overview

This Module of the standard IRDS provides a starter-set of entity-types, relationship-types, relationship-class-types, attribute-types, and attribute-group-types, called the Basic Functional IRD Schema, designed to support intra- and inter-organization communications about information resources.

The IRD Schema descriptors of the Basic Functional IRD Schema exist in the IRD Schema along with the Minimal IRD Schema specified in Module 1, and can be added to the IRD Schema with the use of the IRDS extensibility facility (i.e., the IRD Schema Commands specified in Module 1). There are eight entity-types in the Basic Functional IRD Schema, which conceptually can be grouped into three categories: DATA entity-types, PROCESS entity-types, and EXTERNAL entity-types. These entity-types, by category, are as follows:

### DATA Entity-Types

DOCUMENT  
FILE  
RECORD  
ELEMENT

### PROCESS Entity-Types

SYSTEM  
PROGRAM  
MODULE

### EXTERNAL Entity-Types

USER

The collection of sixty-three relationship-types provided by the Basic Functional IRD Schema is discussed in Subsection 5.2. Most of these relationship-types are grouped into the following seven relationship-class-types:

CALLS  
CONTAINS  
DERIVED-FROM  
GOES-TO  
PROCESSES  
RESPONSIBLE-FOR  
RUNS

The relationship-type/relationship-class-type groupings are presented in Subsection 6.2.

The Basic Functional IRD Schema also consists of twenty-nine attribute-types and three attribute-group-types, which are specified in Subsections 5.4 and 5.5 of this Module, respectively.

## 5 The Basic Functional IRD Schema

The following Subsections provide the entity-types, relationship-types, relationship-class-types, attribute-types, and attribute-group-types which make up the Basic Functional IRD Schema. For each IRD Schema descriptor a name is given, followed by a list of associated meta-attribute-types and their meta-attributes.

The Subsections are organized as follows:

Subsection 5.1 - Entity-Types  
Subsection 5.2 - Relationship-Types  
Subsection 5.3 - Relationship-Class-Types  
Subsection 5.4 - Attribute-Types  
Subsection 5.5 - Attribute-Group-Types

### 5.1 Entity-types in the Basic Functional IRD Schema

To define each entity-type in the Basic Functional IRD Schema, the following information is presented for each entity-type:

- (1) The name of the entity-type, by means of the statement:

Name = <entity-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
META-ENTITY-SUBSTITUTE-NAME  
DATE-TIME-ADDED  
DATE-TIME-LAST-MODIFIED  
LAST-MODIFIED-BY  
NUMBER-OF-TIMES-MODIFIED  
PURPOSE



by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

(3) The meta-attributes of types:

CONNECTABLE  
 ENTITY-CLASS  
 IMPLEMENTATION-LOCK  
 ORIGIN  
 SYSTEM-GENERATED  
 SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1> ... <meta-attribute-type-name-*i*>  
 <meta-attribute>                      <meta-attribute>

**5.1.1 Document**

Name = DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	DATA	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.1.2 Element**

Name = ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	DATA	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.1.3 File**

Name = FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL

DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	DATA	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.1.4 Module**

Name = MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	PROCESS	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.1.5 Program**

Name = PROGRAM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	PROCESS	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.1.6 Record**

Name = RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	DATA	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.1.7 System**

Name = SYSTEM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	PROCESS	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.1.8 User**

Name = USER  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Connectable	Entity Class	Implementation Lock	Origin	System Generated	System Lock
YES	EXTERNAL	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2 Relationship-Types in the Basic Functional IRD Schema**

To define each relationship-type in the Basic Functional IRD Schema, the following information is presented for each relationship-type:

- (1) The name of the relationship-type, by means of the statement:

Name = <relationship-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
 META-ENTITY-SUBSTITUTE-NAME  
 DATE-TIME-ADDED  
 DATE-TIME-LAST-MODIFIED  
 INVERSE-NAME  
 LAST-MODIFIED-BY  
 NUMBER-OF-TIMES-MODIFIED  
 PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>



NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	ON

**5.2.4 Document Derived From Document**

Name = DOCUMENT-DERIVED-FROM-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = DOC-D-FR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PRODUCES-DOCUMENT, DOC-PRD-DOC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.5 Document Derived From File**

Name = DOCUMENT-DERIVED-FROM-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = DOC-D-FR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PRODUCES-DOCUMENT, FIL-PRD-DOC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.6 Document-derived-from-record**

Name = DOCUMENT-DERIVED-FROM-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = DOC-D-FR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-PRODUCES-DOCUMENT, REC-PRD-DOC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.7 Element Contains Element**

Name = ELEMENT-CONTAINS-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = ELE-CON-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-CONTAINED-IN-ELEMENT, ELE-CON-IN-ELE  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	ON

**5.2.8 Element Derived From Document**

Name = ELEMENT-DERIVED-FROM-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = ELE-D-FR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PRODUCES-ELEMENT, DOC-PRD-ELE  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.9 Element Derived From Element**

Name = ELEMENT-DERIVED-FROM-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = ELE-D-FR-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-PRODUCES-ELEMENT, ELE-PRD-ELE  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.10 Element Derived From File**

Name = ELEMENT-DERIVED-FROM-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = ELE-D-FR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PRODUCES-ELEMENT, FIL-PRD-ELE  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin			Lock
OFF	X3.138-1988 (Ch.2)		NO	OFF

### 5.2.11 Element Derived From Record

Name = ELEMENT-DERIVED-FROM-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = ELE-D-FR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-PRODUCES-ELEMENT, REC-PRD-ELE  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin			Lock
OFF	X3.138-1988 (Ch.2)		NO	OFF

### 5.2.12 Element Standard For Element

Name = ELEMENT-STANDARD-FOR-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = ELE-ST-FOR-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-STANDARD-OF-ELEMENT, ELE-ST-OF-ELE  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin			Lock
OFF	X3.138-1988 (Ch.2)		NO	OFF

### 5.2.13 File Contains Document

Name = FILE-CONTAINS-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-CON-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-CONTAINED-IN-FILE, DOC-CON-IN-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System Lock
Lock	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.14 File Contains Element**

Name = FILE-CONTAINS-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-CON-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-CONTAINED-IN-FILE, ELE-CON-IN-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System Lock
Lock	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.15 File Contains File**

Name = FILE-CONTAINS-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-CON-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-CONTAINED-IN-FILE, FIL-CON-IN-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System Lock
Lock	OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.16 File Contains Record**

Name = FILE-CONTAINS-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-CON-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-CONTAINED-IN-FILE, REC-CON-IN-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System Lock
Lock	OFF	X3.138-1988 (Ch.2)	NO	ON



**5.2.17 File Derived From Document**

Name = FILE-DERIVED-FROM-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-D-FR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PRODUCES-FILE, DOC-PRD-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			System
Lock	Origin	Sequenced	Lock
OFF	X3.138-1988 (Ch.2)	NO	OFF

**5.2.18 File Derived From File**

Name = FILE-DERIVED-FROM-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-D-FR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PRODUCES-FILE, FIL-PRD-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			System
Lock	Origin	Sequenced	Lock
OFF	X3.138-1988 (Ch.2)	NO	OFF

**5.2.19 File Has Access Key Element**

Name = FILE-HAS-ACCESS-KEY-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-H-A-K-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-ACCESS-KEY-OF-FILE, ELE-A-K-OF-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			System
Lock	Origin	Sequenced	Lock
OFF	X3.138-1988 (Ch.2)	NO	OFF

**5.2.20 File Has Sort Key Element**

Name = FILE-HAS-SORT-KEY-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = FIL-H-S-K-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-SORT-KEY-OF-FILE, ELE-S-K-OF-FIL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		OFF

**5.2.21 Module Calls Module**

Name = MODULE-CALLS-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL-CLS-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-CALLED-BY-MODULE, MDL-CLD-BY-MDL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		OFF

**5.2.22 Module Contains Module**

Name = MODULE-CONTAINS-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL-CON-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-CONTAINED-IN-MODULE, MDL-CON-IN-MDL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		ON

**5.2.23 Module Goes To Module**

Name = MODULE-GOES-TO-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL-TO-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-COMES-FROM-MODULE, MDL-FR-MDL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock OFF

**5.2.24 Module Processes Document**

Name = MODULE-PROCESSES-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL-PR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PROCESSED-BY-MODULE, DOC-PR-BY-MDL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock ON

**5.2.25 Module Processes Element**

Name = MODULE-PROCESSES-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL-PR-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-PROCESSED-BY-MODULE, ELE-PR-BY-MDL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock ON

**5.2.26 Module Processes File**

Name = MODULE-PROCESSES-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL-PR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PROCESSED-BY-MODULE, FIL-PR-BY-MDL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock ON

**5.2.27 Module Processes Record**

Name = MODULE-PROCESSES-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = MDL-PR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-PROCESSED-BY-MODULE, REC-PR-BY-MDL  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		ON

**5.2.28 Program Calls Module**

Name = PROGRAM-CALLS-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-CLS-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-CALLED-BY-PROGRAM, MDL-CLD-BY-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		OFF

**5.2.29 Program Calls Program**

Name = PROGRAM-CALLS-PROGRAM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-CLS-PGM  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PROGRAM-CALLED-BY-PROGRAM, PGM-CLD-BY-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		OFF

**5.2.30 Program Contains Module**

Name = PROGRAM-CONTAINS-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-CON-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-CONTAINED-IN-PROGRAM, MDL-CON-IN-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		ON

**5.2.31 Program Contains Program**

Name = PROGRAM-CONTAINS-PROGRAM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-CON-PGM  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PROGRAM-CONTAINED-IN-PROGRAM, PGM-CON-IN-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		ON

**5.2.32 Program Goes To Program**

Name = PROGRAM-GOES-TO-PROGRAM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-TO-PGM  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PROGRAM-COMES-FROM-PROGRAM, PGM-FR-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		OFF

**5.2.33 Program Processes Document**

Name = PROGRAM-PROCESSES-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-PR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PROCESSED-BY-PROGRAM, DOC-PR-BY-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System Lock
Lock OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.34 Program Processes Element**

Name = PROGRAM-PROCESSES-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-PR-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-PROCESSED-BY-PROGRAM, ELE-PR-BY-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System Lock
Lock OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.35 Program Processes File**

Name = PROGRAM-PROCESSES-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-PR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PROCESSED-BY-PROGRAM, FIL-PR-BY-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System Lock
Lock OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.36 Program Processes Record**

Name = PROGRAM-PROCESSES-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PGM-PR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-PROCESSED-BY-PROGRAM, REC-PR-BY-PGM  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System Lock
Lock OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.37 Record Contains Element**

Name = RECORD-CONTAINS-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = REC-CON-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-CONTAINED-IN-RECORD, ELE-CON-N-REC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		ON

**5.2.38 Record Contains Record**

Name = RECORD-CONTAINS-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = REC-CON-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-CONTAINED-IN-RECORD, REC-CON-IN-REC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		ON

**5.2.39 Record Derived From Document**

Name = RECORD-DERIVED-FROM-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = REC-D-FR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PRODUCES-RECORD, DOC-PRD-REC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock	Origin	Sequenced		Lock
OFF	X3.138-1988 (Ch.2)	NO		OFF

**5.2.40 Record Derived From File**

Name = RECORD-DERIVED-FROM-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = REC-D-FR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PRODUCES-RECORD, FIL-PRD-REC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin		Lock	
OFF	X3.138-1988 (Ch.2)		NO	OFF

**5.2.41 Record Derived From Record**

Name = RECORD-DERIVED-FROM-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = REC-D-FR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-PRODUCES-RECORD, REC-PRD-REC  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin		Lock	
OFF	X3.138-1988 (Ch.2)		NO	OFF

**5.2.42 System Contains Module**

Name = SYSTEM-CONTAINS-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-CON-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-CONTAINED-IN-SYSTEM, MDL-CON-IN-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin		Lock	
OFF	X3.138-1988 (Ch.2)		NO	ON

**5.2.43 System Contains Program**

Name = SYSTEM-CONTAINS-PROGRAM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-CON-PGM  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PROGRAM-CONTAINED-IN-SYSTEM, PGM-CON-IN-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/



Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock ON

**5.2.44 System Contains System**

Name = SYSTEM-CONTAINS-SYSTEM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-CON-SYS  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = SYSTEM-CONTAINED-IN-SYSTEM, SYS-CON-IN-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock ON

**5.2.45 System Goes To System**

Name = SYSTEM-GOES-TO-SYSTEM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-TO-SYS  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = SYSTEM-COMES-FROM-SYSTEM, SYS-FR-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock OFF

**5.2.46 System Processes Document**

Name = SYSTEM-PROCESSES-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-PR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PROCESSED-BY-SYSTEM, DOC-PR-BY-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock OFF	X3.138-1988 (Ch.2)	NO	Lock ON

**5.2.47 System Processes Element**

Name = SYSTEM-PROCESSES-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-PR-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-PROCESSED-BY-SYSTEM, ELE-PR-BY-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	ON

**5.2.48 System Processes File**

Name = SYSTEM-PROCESSES-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-PR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PROCESSED-BY-SYSTEM, FIL-PR-BY-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	ON

**5.2.49 System Processes Record**

Name = SYSTEM-PROCESSES-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = SYS-PR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-PROCESSED-BY-SYSTEM, REC-PR-BY-SYS  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	ON

**5.2.50 User Processes Document**

Name = USER-PROCESSES-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-PR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-PROCESSED-BY-USER, DOC-PR-BY-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock			Lock
OFF	X3.138-1988 (Ch.2)	NO	ON

### 5.2.51 User Processes Element

Name = USER-PROCESSES-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-PR-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-PROCESSED-BY-USER, ELE-PR-BY-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock			Lock
OFF	X3.138-1988 (Ch.2)	NO	ON

### 5.2.52 User Processes File

Name = USER-PROCESSES-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-PR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-PROCESSED-BY-USER, FIL-PR-BY-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	Sequenced	System
Lock			Lock
OFF	X3.138-1988 (Ch.2)	NO	ON

### 5.2.53 User Processes Record

Name = USER-PROCESSES-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-PR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-PROCESSED-BY-USER, REC-PR-BY-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			System
Lock	Origin	Sequenced	Lock
OFF	X3.138-1988 (Ch.2)	NO	ON

**5.2.54 User Responsible For Document**

Name = USER-RESPONSIBLE-FOR-DOCUMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-R-FOR-DOC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = DOCUMENT-RESPONSIBILITY-OF-USER, DOC-R-OF-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			System
Lock	Origin	Sequenced	Lock
OFF	X3.138-1988 (Ch.2)	NO	OFF

**5.2.55 User Responsible For Element**

Name = USER-RESPONSIBLE-FOR-ELEMENT  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-R-FOR-ELE  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = ELEMENT-RESPONSIBILITY-OF-USER, ELE-R-OF-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			System
Lock	Origin	Sequenced	Lock
OFF	X3.138-1988 (Ch.2)	NO	OFF

**5.2.56 User Responsible For File**

Name = USER-RESPONSIBLE-FOR-FILE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-R-FOR-FIL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = FILE-RESPONSIBILITY-OF-USER, FIL-R-OF-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			System
Lock	Origin	Sequenced	Lock
OFF	X3.138-1988 (Ch.2)	NO	OFF

**5.2.57 User Responsible For Module**

Name = USER-RESPONSIBLE-FOR-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-R-FOR-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-RESPONSIBILITY-OF-USER, MDL-R-OF-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin		Lock	
OFF	X3.138-1988 (Ch.2)		NO	OFF

**5.2.58 User Responsible For Program**

Name = USER-RESPONSIBLE-FOR-PROGRAM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-R-FOR-PGM  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PROGRAM-RESPONSIBILITY-OF-USER, PGM-R-OF-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin		Lock	
OFF	X3.138-1988 (Ch.2)		NO	OFF

**5.2.59 User Responsible For Record**

Name = USER-RESPONSIBLE-FOR-RECORD  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-R-FOR-REC  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RECORD-RESPONSIBILITY-OF-USER, REC-R-OF-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation			Sequenced	System
Lock	Origin		Lock	
OFF	X3.138-1988 (Ch.2)		NO	OFF

**5.2.60 User Responsible For System**

Name = USER-RESPONSIBLE-FOR-SYSTEM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-R-FOR-SYS  
 DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = SYSTEM-RESPONSIBILITY-OF-USER, SYS-R-OF-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.61 User Runs Module**

Name = USER-RUNS-MODULE  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-RUNS-MDL  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = MODULE-RUN-BY-USER, MDL-RUN-BY-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.62 User Runs Program**

Name = USER-RUNS-PROGRAM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-RUNS-PGM  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PROGRAM-RUN-BY-USER, PGM-RUN-BY-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation		Origin	Sequenced	System
Lock				Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

**5.2.63 User Runs System**

Name = USER-RUNS-SYSTEM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = USR-RUNS-SYS  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = SYSTEM-RUN-BY-USER, SYS-RUN-BY-USR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation				System
Lock		Origin	Sequenced	Lock
OFF		X3.138-1988 (Ch.2)	NO	OFF

### 5.3 Relationship-Class-Types in the Basic Functional IRD Schema

To define each relationship-class-type in the Basic Functional IRD Schema, the following information is presented for each relationship-class-type:

- (1) The name of the relationship-class-type, by means of the statement:

Name = <relationship-class-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
 META-ENTITY-SUBSTITUTE-NAME  
 DATE-TIME-ADDED  
 DATE-TIME-LAST-MODIFIED  
 INVERSE-NAME  
 LAST-MODIFIED-BY  
 NUMBER-OF-TIMES-MODIFIED  
 PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
 ORIGIN  
 SYSTEM-LOCK

by means of a table of the form:

<meta-attribute-type-name-1> ... <meta-attribute-type-name-i>  
 <meta-attribute>                      <meta-attribute>

#### 5.3.1 Calls

Name = CALLS  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = CLS  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = CALLED-BY, CLD-BY  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	System
Lock ON	X3.138-1988 (Ch.2)	Lock OFF

**5.3.2 Contains**

Name = CONTAINS  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = CON  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = CONTAINED-IN, CON-IN  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	System
Lock OFF	X3.138-1988 (Ch.2)	Lock OFF

**5.3.3 Derived From**

Name = DERIVED-FROM  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = D-FR  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PRODUCES, PRD  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	System
Lock OFF	X3.138-1988 (Ch.2)	Lock OFF

**5.3.4 Goes To**

Name = GOES-TO  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = TO  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = COMES-FROM, FR  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	System
Lock OFF	X3.138-1988 (Ch.2)	Lock OFF



**5.3.5 Processes**

Name = PROCESSES  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = PR  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = PROCESSED-BY, PR-BY  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	System
Lock		Lock
OFF	X3.138-1988 (Ch.2)	OFF

**5.3.6 Responsible For**

Name = RESPONSIBLE-FOR  
 ADDED-BY = /\* implementor defined \*/  
 META-ENTITY-SUBSTITUTE-NAME = R-FOR  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RESPONSIBILITY-OF, R-OF  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	System
Lock		Lock
OFF	X3.138-1988 (Ch.2)	OFF

**5.3.7 Runs**

Name = RUNS  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 INVERSE-NAME = RUN-BY  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = /\* implementor defined \*/

Implementation	Origin	System
Lock		Lock
OFF	X3.138-1988 (Ch.2)	OFF

**5.4 Attribute-Types in the Basic Functional IRD Schema**

To define each attribute-type in the Basic Functional IRD Schema, the following information is presented for each attribute-type:



**5.4.2 Allowable Value**

Name = ALLOWABLE-VALUE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To record in the IRD the allowable value(s) that can be taken on by a real world element."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.3 Alternate Name**

Name = ALTERNATE-NAME  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "An attribute-type of the IDENTIFICATION-NAMES attribute-group-type used to record names in the IRD (other than the access-name or the descriptive-name) by which an entity is known."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
YES	STRING	OFF	X3.138-1988 (Ch.2)	ON

**5.4.4 Alternate Name Context**

Name = ALTERNATE-NAME-CONTEXT  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "An attribute-type of the IDENTIFICATION-NAMES attribute-group-type used to document in the IRD the context or environment in which an alternate name for an entity exists."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
YES	STRING	OFF	X3.138-1988 (Ch.2)	ON

**5.4.5 Classification**

Name = CLASSIFICATION  
 ADDED-BY = /\* implementor defined \*/

DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To provide an installation with the capability to specify the use of a classification scheme for entities in the IRD."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	ON

**5.4.6 Code List Location**

Name = CODE-LIST-LOCATION  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To record in the IRD the location(s), outside the IRD at which a list of codes for an element is located."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.7 Comments**

Name = COMMENTS  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To document clarifying information about an entity."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
YES	TEXT	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.8 Data Class**

Name = DATA-CLASS  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To document in the IRD the general default character of an instance of an ELEMENT."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.9 Data Type**

Name = DATA-TYPE

ADDED-BY = /\* implementor defined \*/

DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/

LAST-MODIFIED-BY = /\* implementor defined \*/

NUMBER-OF-INSTANCES = Zero

NUMBER-OF-TIMES-MODIFIED = Zero

PURPOSE = "To document in the IRD the data type of an instance of an ELEMENT.  
Possible values are BIT-STRING, CHARACTER-STRING, FIXED-POINT, and  
FLOAT."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.10 Description**

Name = DESCRIPTION

ADDED-BY = /\* implementor defined \*/

DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/

LAST-MODIFIED-BY = /\* implementor defined \*/

NUMBER-OF-INSTANCES = Zero

NUMBER-OF-TIMES-MODIFIED = Zero

PURPOSE = "To describe or define an IRD descriptor."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
YES	TEXT	OFF	X3.138-1988 (Ch.2)	ON

**5.4.11 Document Category**

Name = DOCUMENT-CATEGORY

ADDED-BY = /\* implementor defined \*/

DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/

LAST-MODIFIED-BY = /\* implementor defined \*/

NUMBER-OF-INSTANCES = Zero

NUMBER-OF-TIMES-MODIFIED = Zero

PURPOSE = "To allow an installation to establish a classification scheme for documents."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.12 Duration Type**

Name = DURATION-TYPE

ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "Used in conjunction with the DURATION-VALUE attribute-type to provide the units associated with the numeric value assigned to DURATION-VALUE in an attribute-group of type DURATION."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.13 Duration Value**

Name = DURATION-VALUE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "Used in conjunction with the DURATION-TYPE attribute-type to document a magnitude of time in an attribute-group of type DURATION."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	REAL	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.14 External Security**

Name = EXTERNAL-SECURITY  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To document the security requirements of entity instances in the real world."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.15 Frequency**

Name = FREQUENCY  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero

NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To document the expected frequency at which a real-world event occurs.  
 This attribute-type is associated with relationship-types of relationship-class-  
 types PROCESSES and RUNS."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.16 High Of Range**

Name = HIGH-OF-RANGE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "An attribute-type of the ALLOWABLE-RANGE attribute-group-type, used to  
 record in the IRD the high value of a range of values that can be taken on  
 by a real world ELEMENT."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.17 Internal Format**

Name = INTERNAL-FORMAT  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To allow the internal representation of an ELEMENT to be defined.  
 Possible values are ASCII, EBCDIC, BINARY, DECIMAL, PACKED,  
 COMPRESSED, and DOUBLE."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	ON

**5.4.18 Justification**

Name = JUSTIFICATION  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "JUSTIFICATION describes the positioning of characters in an ELEMENT  
 with data-type CHARACTER-STRING. Valid values are RIGHT and LEFT."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	ON

**5.4.19 Length**

Name = LENGTH  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "For an ELEMENT, specifies the number of characters for a data-type of CHARACTER-STRING, the number of bits for a data-type of BIT-STRING, or the number of digits for a data-type of FIXED-POINT or FLOAT."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	INTEGER	OFF	X3.138-1988 (Ch.2)	ON

**5.4.20 Location**

Name = LOCATION  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To document the place(s) within an organization where an entity can be found."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.21 Low Of Range**

Name = LOW-OF-RANGE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "An attribute-type of the ALLOWABLE-RANGE attribute-group-type, used to record in the IRD the low value of a range of values that can be taken on by a real world ELEMENT."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF



**5.4.22 Number Of Lines Of Code**

Name = NUMBER-OF-LINES-OF-CODE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To document the number of lines of language statements which are associated with a real world PROGRAM or MODULE."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	INTEGER	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.23 Number Of Records**

Name = NUMBER-OF-RECORDS  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "The number of logical records expected to exist in the real world FILE instance."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	INTEGER	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.24 Precision**

Name = PRECISION  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "A positive integer that determines the number of significant decimal digits to the right of the decimal point in an ELEMENT of data type FIXED-POINT or FLOAT."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	INTEGER	OFF	X3.138-1988 (Ch.2)	ON

**5.4.25 Record Category**

Name = RECORD-CATEGORY  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To allow an installation to establish a classification scheme for RECORDS."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.26 Relative Position**

Name = RELATIVE-POSITION  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To identify the relative position of an ELEMENT in a RECORD."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	INTEGER	OFF	X3.138-1988 (Ch.2)	OFF

**5.4.27 Scale**

Name = SCALE  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "A signed integer that determines placement of the decimal point in an ELEMENT of data-type FIXED-POINT or FLOAT."

Common To		Implementation		System
Entity-Types	Format	Lock	Origin	Lock
NO	INTEGER	OFF	X3.138-1988 (Ch.2)	ON

**5.4.28 System Category**

Name = SYSTEM-CATEGORY  
 ADDED-BY = /\* implementor defined \*/  
 DATE-TIME-ADDED = /\* implementor defined \*/  
 DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/  
 LAST-MODIFIED-BY = /\* implementor defined \*/  
 NUMBER-OF-INSTANCES = Zero  
 NUMBER-OF-TIMES-MODIFIED = Zero  
 PURPOSE = "To allow an installation to establish a classification scheme for SYSTEMS."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

#### 5.4.29 Usage

Name = USAGE

ADDED-BY = /\* implementor defined \*/

DATE-TIME-ADDED = /\* implementor defined \*/

DATE-TIME-LAST-MODIFIED = /\* implementor defined \*/

LAST-MODIFIED-BY = /\* implementor defined \*/

NUMBER-OF-INSTANCES = Zero

NUMBER-OF-TIMES-MODIFIED = Zero

PURPOSE = "To allow the usage of an ELEMENT to be defined. Possible values are MONEY, DATE, TIME, POINTER, and BOOLEAN."

Common To Entity-Types	Format	Implementation Lock	Origin	System Lock
NO	STRING	OFF	X3.138-1988 (Ch.2)	OFF

#### 5.5 Attribute-Group-Types in the Basic Functional IRD Schema

To define each attribute-group-type in the Basic Functional IRD Schema, the following information is presented for each attribute-group-type:

- (1) The name of the attribute-group-type, by means of the statement:

Name = <attribute-group-type-name>

- (2) The meta-attributes of types:

ADDED-BY  
DATE-TIME-ADDED  
DATE-TIME-LAST-MODIFIED  
LAST-MODIFIED-BY  
NUMBER-OF-INSTANCES  
NUMBER-OF-TIMES-MODIFIED  
PURPOSE

by means of statements of the form:

<meta-attribute-type-name> = <meta-attribute>

- (3) The meta-attributes of types:

IMPLEMENTATION-LOCK  
ORIGIN  
SIGNIFICANT-ATTRIBUTES  
SYSTEM-LOCK



## 6 Associations between Basic Functional IRD Schema Descriptors

The associations between these descriptors are expressed in terms of the meta-relationships that exist in the Basic Functional IRD Schema. Applicable meta-attributes are specified.

### 6.1 Entity-types Associated with Relationship-Types

The Basic Functional IRD Schema shall contain meta-relationships of type RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE that associate relationship-types and the entity-types which are members of each relationship-type according to the following table. Table 1 shows the relationship-types which are associated with the following entity-types:

DOCUMENT  
ELEMENT  
FILE  
MODULE  
PROGRAM  
RECORD  
SYSTEM  
USER

The legend used in the table is as follows:

- 1 denotes that the entity-type is the first member of the relationship-type (i.e., that the meta-attribute of type POSITION of the meta-relationship between the relationship-type and entity-type is 1).
- 2 denotes that the entity-type is the second member of the relationship-type (i.e., that the meta-attribute of type POSITION of the meta-relationship between the relationship-type and entity-type is 2).
- R denotes that the entity-type is both the first and the second member of the relationship-type (i.e., that there exist two meta-relationships between the entity-type and relationship-type, one where the meta-attribute of type POSITION is 1, and the other where the meta-attribute of this type is 2).

### 6.2 Relationship-Types Associated with Relationship-Class-Types

The Basic Functional IRD Schema shall contain the following meta-relationships of type RELATIONSHIP-TYPE-MEMBER-OF-RELATIONSHIP-CLASS-TYPE:

The relationship-types:

MODULE - CALLS - MODULE  
PROGRAM - CALLS - MODULE  
PROGRAM - CALLS - PROGRAM

and the relationship-class-type

CALLS .

**Table 1 (Page 1 of 3)  
Association of Entity-Types and Relationship-Types**

Relationship-type	Entity-type							
	DOC	ELE	FIL	MDL	PGR	REC	SYS	USR
DOCUMENT-CONTAINS-DOCUMENT	R							
DOCUMENT-CONTAINS-ELEMENT	1	2						
DOCUMENT-CONTAINS-RECORD	1					2		
DOCUMENT-DERIVED-FROM-DOCUMENT	R							
DOCUMENT-DERIVED-FROM-FILE	1		2					
DOCUMENT-DERIVED-FROM-RECORD	1					2		
ELEMENT-CONTAINS-ELEMENT		R						
ELEMENT-DERIVED-FROM-DOCUMENT	2	1						
ELEMENT-DERIVED-FROM-ELEMENT		R						
ELEMENT-DERIVED-FROM-FILE		1	2					
ELEMENT-DERIVED-FROM-RECORD		1				2		
ELEMENT-STANDARD-FOR-ELEMENT		R						
FILE-CONTAINS-DOCUMENT	2		1					
FILE-CONTAINS-ELEMENT		2	1					
FILE-CONTAINS-FILE			R					
FILE-CONTAINS-RECORD			1			2		
FILE-DERIVED-FROM-DOCUMENT	2		1					
FILE-DERIVED-FROM-FILE			R					
FILE-HAS-ACCESS-KEY-ELEMENT		2	1					
FILE-HAS-SORT-KEY-ELEMENT		2	1					
MODULE-CALLS-MODULE				R				
MODULE-CONTAINS-MODULE				R				
Relationship-type	DOC	ELE	FIL	MDL	PGM	REC	SYS	USR

**Table 1 (Page 2 of 3)**  
**Association of Entity-Types and Relationship-Types**

Relationship-type	Entity-type							
	DOC	ELE	FIL	MDL	PGR	REC	SYS	USR
MODULE-GOES-TO-MODULE				R				
MODULE-PROCESSES-DOCUMENT	2			1				
MODULE-PROCESSES-ELEMENT		2		1				
MODULE-PROCESSES-FILE			2	1				
MODULE-PROCESSES-RECORD				1		2		
PROGRAM-CALLS-MODULE				2	1			
PROGRAM-CALLS-PROGRAM					R			
PROGRAM-CONTAINS-MODULE				2	1			
PROGRAM-CONTAINS-PROGRAM					R			
PROGRAM-GOES-TO-PROGRAM					R			
PROGRAM-PROCESSES-DOCUMENT	2				1			
PROGRAM-PROCESSES-ELEMENT		2			1			
PROGRAM-PROCESSES-FILE			2		1			
PROGRAM-PROCESSES-RECORD					1	2		
RECORD-CONTAINS-ELEMENT		2				1		
RECORD-CONTAINS-RECORD						R		
RECORD-DERIVED-FROM-DOCUMENT	2					1		
RECORD-DERIVED-FROM-FILE			2			1		
RECORD-DERIVED-FROM-RECORD						R		
SYSTEM-CONTAINS-MODULE				2			1	
SYSTEM-CONTAINS-PROGRAM					2		1	
SYSTEM-CONTAINS-SYSTEM							R	
Relationship-type	DOC	ELE	FIL	MDL	PGM	REC	SYS	USR

**Table 1 (Page 3 of 3)  
Association of Entity-Types and Relationship-Types**

Relationship-type	Entity-type							
	DOC	ELE	FIL	MDL	PGR	REC	SYS	USR
SYSTEM-GOES-TO-SYSTEM							R	
SYSTEM-PROCESSES-DOCUMENT	2						1	
SYSTEM-PROCESSES-ELEMENT		2					1	
SYSTEM-PROCESSES-FILE			2				1	
SYSTEM-PROCESSES-RECORD						2	1	
USER-PROCESSES-DOCUMENT	2							1
USER-PROCESSES-ELEMENT		2						1
USER-PROCESSES-FILE			2					1
USER-PROCESSES-RECORD						2		1
USER-RESPONSIBLE-FOR-DOCUMENT	2							1
USER-RESPONSIBLE-FOR-ELEMENT		2						1
USER-RESPONSIBLE-FOR-FILE			2					1
USER-RESPONSIBLE-FOR-MODULE				2				1
USER-RESPONSIBLE-FOR-PROGRAM					2			1
USER-RESPONSIBLE-FOR-RECORD						2		1
USER-RESPONSIBLE-FOR-SYSTEM							2	1
USER-RUNS-MODULE				2				1
USER-RUNS-PROGRAM					2			1
USER-RUNS-SYSTEM							2	1
Relationship-type	DOC	ELE	FIL	MDL	PGM	REC	SYS	USR



The relationship-types:

DOCUMENT-CONTAINS-DOCUMENT  
DOCUMENT-CONTAINS-ELEMENT  
DOCUMENT-CONTAINS-RECORD  
ELEMENT-CONTAINS-ELEMENT  
FILE-CONTAINS-DOCUMENT  
FILE-CONTAINS-ELEMENT  
FILE-CONTAINS-FILE  
FILE-CONTAINS-RECORD  
MODULE-CONTAINS-MODULE  
PROGRAM-CONTAINS-MODULE  
PROGRAM-CONTAINS-PROGRAM  
RECORD-CONTAINS-ELEMENT  
RECORD-CONTAINS-RECORD  
SYSTEM-CONTAINS-MODULE  
SYSTEM-CONTAINS-PROGRAM  
SYSTEM-CONTAINS-SYSTEM

and the relationship-class-type

CONTAINS.

The relationship-types:

DOCUMENT-DERIVED-FROM-DOCUMENT  
DOCUMENT-DERIVED-FROM-FILE  
DOCUMENT-DERIVED-FROM-RECORD  
ELEMENT-DERIVED-FROM-DOCUMENT  
ELEMENT-DERIVED-FROM-ELEMENT  
ELEMENT-DERIVED-FROM-FILE  
ELEMENT-DERIVED-FROM-RECORD  
FILE-DERIVED-FROM-DOCUMENT  
FILE-DERIVED-FROM-FILE  
RECORD-DERIVED-FROM-DOCUMENT  
RECORD-DERIVED-FROM-FILE  
RECORD-DERIVED-FROM-RECORD

and the relationship-class-type

DERIVED-FROM.

The relationship-types:

MODULE-GOES-TO-MODULE  
PROGRAM-GOES-TO-PROGRAM  
SYSTEM-GOES-TO-SYSTEM

and the relationship-class-type

GOES-TO.

The relationship-types:

MODULE-PROCESSES-DOCUMENT  
MODULE-PROCESSES-ELEMENT  
MODULE-PROCESSES-FILE  
MODULE-PROCESSES-RECORD  
PROGRAM-PROCESSES-DOCUMENT  
PROGRAM-PROCESSES-ELEMENT  
PROGRAM-PROCESSES-FILE  
PROGRAM-PROCESSES-RECORD  
SYSTEM-PROCESSES-DOCUMENT  
SYSTEM-PROCESSES-ELEMENT  
SYSTEM-PROCESSES-FILE  
SYSTEM-PROCESSES-RECORD  
USER-PROCESSES-DOCUMENT  
USER-PROCESSES-ELEMENT  
USER-PROCESSES-FILE  
USER-PROCESSES-RECORD

and the relationship-class-type

PROCESSES.

The relationship-types:

USER-RESPONSIBLE-FOR-DOCUMENT  
USER-RESPONSIBLE-FOR-ELEMENT  
USER-RESPONSIBLE-FOR-FILE  
USER-RESPONSIBLE-FOR-MODULE  
USER-RESPONSIBLE-FOR-PROGRAM  
USER-RESPONSIBLE-FOR-RECORD  
USER-RESPONSIBLE-FOR-SYSTEM

and the relationship-class-type

RESPONSIBLE-FOR.

The relationship-types:

USER-RUNS-MODULE  
USER-RUNS-PROGRAM  
USER-RUNS-SYSTEM

and the relationship-class-type

RUNS.

### **6.3 Attribute-Types Associated with Entity-Types**

The Basic Functional IRD Schema shall contain meta-relationships that associate entity-types and attribute-types according to Table 2. This Table shows the attribute-types which are associated with the following entity-types:

DOCUMENT  
 ELEMENT  
 FILE  
 MODULE  
 PROGRAM  
 RECORD  
 SYSTEM  
 USER

Those attribute-type names which are designated with a suffix of "(M1)" are attribute-types which are defined in Module 1 of this standard, i.e., which are part of the Minimal IRD Schema.

At the intersection of a row and column:

Y denotes that the meta-attribute of type SINGULAR of the meta-relationship is YES.

N denotes that the meta-attribute of type SINGULAR of the meta-relationship is NO.

#### 6.4 Attribute-Group-Types Associated with Entity-Types

The Basic Functional IRD Schema shall contain meta-relationships that associate entity-types and attribute-group-types according to Table 3. The legend used is:

At the intersection of a row and column:

Y denotes that the meta-attribute of type SINGULAR of the meta-relationship is YES.

N denotes that the meta-attribute of type SINGULAR of the meta-relationship is NO.

Those attribute-group-type names which are designated with a suffix of "(M1)" are attribute-group-types which are defined in Module 1 of this standard, i.e., which are part of the Minimal IRD Schema.

#### 6.5 Attribute-Types Associated with Relationship-Types

The Basic Functional IRD Schema shall contain meta-relationships of type RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-TYPE as follows:

For the relationship-types:

SYSTEM-PROCESSES-FILE,  
 PROGRAM-PROCESSES-FILE, and  
 MODULE-PROCESSES-FILE,

the attribute-type ACCESS-METHOD, where the meta-attribute of type SINGULAR is YES.

For all the relationship-types associated with the relationship-class-types PROCESSES and RUNS:

The attribute-type FREQUENCY, where the meta-attribute of type SINGULAR is YES.

**Table 2**  
**Association of Entity-Types and Attribute-Types**

Attribute-type	Entity-type							
	DOC	ELE	FIL	MDL	PGR	REC	SYS	USR
ADDED-BY (M1)	Y	Y	Y	Y	Y	Y	Y	Y
ALLOWABLE-VALUE		N						
CLASSIFICATION	N	N	N	N	N	N	N	N
CODE-LIST-LOCATION		N						
COMMENTS	Y	Y	Y	Y	Y	Y	Y	Y
DATA-CLASS		Y						
DATA-TYPE		Y						
DESCRIPTION	Y	Y	Y	Y	Y	Y	Y	Y
DOCUMENT-CATEGORY	Y							
EXTERNAL-SECURITY	Y	Y	Y	Y	Y	Y	Y	Y
INTERNAL-FORMAT		N						
JUSTIFICATION		Y						
LAST-MODIFIED-BY (M1)	Y	Y	Y	Y	Y	Y	Y	Y
LENGTH		Y						
LOCATION	N		N	N	N		N	N
NUMBER-OF-LINES-OF-CODE				Y	Y			
NUMBER-OF-TIMES-MODIFIED (M1)	Y	Y	Y	Y	Y	Y	Y	Y
NUMBER-OF-RECORDS			Y					
PRECISION		Y						
RECORD-CATEGORY						Y		
SCALE		Y						
SYSTEM-CATEGORY							Y	
USAGE		N						

**Table 3**  
**Association of Entity-Types and Attribute-Group-Types**

Attribute-group-type	Entity-type							
	DOC	ELE	FIL	MDL	PGR	REC	SYS	USR
ALLOWABLE-RANGE		N						
DATE-TIME-ADDED (M1)	Y	Y	Y	Y	Y	Y	Y	Y
DATE-TIME-LAST-MODIFIED (M1)	Y	Y	Y	Y	Y	Y	Y	Y
DURATION				Y	Y		Y	
IDENTIFICATION-NAMES	N	N	N	N	N	N	N	N

For the relationship-type RECORD-CONTAINS-ELEMENT:

The attribute-type RELATIVE-POSITION, where the meta-attribute of type SINGULAR is YES.

### 6.6 Attribute-Group-Types Associated with Relationship-Types

The Basic Functional IRD Schema does not contain any meta-relationships of type RELATIONSHIP-TYPE-CONTAINS-ATTRIBUTE-GROUP-TYPE.

### 6.7 Attribute-Types Associated with Attribute-Group-Types

The Basic Functional IRD Schema shall contain meta-relationships of type ATTRIBUTE-GROUP-TYPE-CONTAINS-ATTRIBUTE-TYPE and meta-attributes of type GROUP-POSITION as follows:

For the attribute-group-type ALLOWABLE-RANGE:

The attribute-type LOW-OF-RANGE with meta-attribute of type GROUP-POSITION equal to 1;

The attribute-type HIGH-OF-RANGE with meta-attribute of type GROUP-POSITION equal to 2.

For the attribute-group-type DURATION:

The attribute-type DURATION-VALUE with meta-attribute of type GROUP-POSITION equal to 1;

The attribute-type DURATION-TYPE with meta-attribute of type GROUP-POSITION equal to 2.

For the attribute-group-type IDENTIFICATION-NAMES:

The attribute-type ALTERNATE-NAME with meta-attribute of type GROUP-POSITION equal to 1;

The attribute-type ALTERNATE-NAME-CONTEXT with meta-attribute of type GROUP-POSITION equal to 2.

**Chapter 3****Module 3 - IRDS Security****CONTENTS**

<b>SECTION</b>	<b>PAGE</b>
1 Scope, Purpose, and Application . . . . .	3-1
1.1 Scope . . . . .	3-1
1.2 Purpose . . . . .	3-1
1.3 Architecture of the IRDS Security Facility . . . . .	3-1
1.4 Conformance . . . . .	3-3
1.5 Organization . . . . .	3-3
1.6 Notation . . . . .	3-4
2 References . . . . .	3-6
3 Definitions . . . . .	3-6
4 Module Overview . . . . .	3-8
4.1 Description of Security Facilities . . . . .	3-8
4.1.1 The Global Security Facility . . . . .	3-10
4.1.2 The Entity-level Security Facility . . . . .	3-10
4.2 Security Descriptors . . . . .	3-11
5 Basic Language Elements . . . . .	3-15
5.1 Function . . . . .	3-15
5.2 Format . . . . .	3-15
5.3 Syntax Rules . . . . .	3-16
5.4 General Rules . . . . .	3-16
5.5 Error and Warning Conditions . . . . .	3-16
6 IRDS Security Commands . . . . .	3-16
6.1 Add Security Command . . . . .	3-17
6.2 Modify Security Command . . . . .	3-20
6.3 Delete Security Command . . . . .	3-22
6.4 Add Access-Key Command . . . . .	3-25
6.5 Delete Access-Key Command . . . . .	3-27
7 IRDS Security Command-Clauses . . . . .	3-29
7.1 Entities In Command-Clause . . . . .	3-29
7.2 From Controller Command-Clause . . . . .	3-30
7.3 To Controller Command-Clause . . . . .	3-30
7.4 Related Controllers Command-Clause . . . . .	3-31
7.5 Controller-List Command-Clause . . . . .	3-32
7.6 IRD-Views List Command-Clause . . . . .	3-32
7.7 For All Controllers Command-Clause . . . . .	3-33
7.8 Assign Security Command-Clause . . . . .	3-33
7.9 Exclude Security Command-Clause . . . . .	3-34

SECTION	PAGE	
7.10	IRD-Views In Command-Clause . . . . .	3-34
7.11	For Controllers In Command-Clause . . . . .	3-35
8	Modifications to Module 1 Commands and Panel Trees . . . . .	3-36
8.1	Modifications to IRD Schema Commands . . . . .	3-38
8.1.1	Modifications to IRD Schema Maintenance Commands . . . . .	3-42
8.1.1.1	Modifications to Add Meta-Entity Command . . . . .	3-42
8.1.1.2	Modifications to Modify Meta-Entity Command . . . . .	3-42
8.1.1.3	Modifications to Delete Meta-Entity Command . . . . .	3-43
8.1.1.4	Modifications to Add Meta-Relationship Command . . . . .	3-44
8.1.1.5	Modifications to Modify Meta-Relationship Command . . . . .	3-44
8.1.1.6	Modifications to Delete Meta-relationship Command . . . . .	3-45
8.1.1.7	Modifications to Modify Meta-Entity Access-Name Command . . . . .	3-45
8.1.1.8	Modifications to Modify Meta-Entity Descriptive-Name Command . . . . .	3-46
8.1.1.9	Modifications to Modify Meta-Entity Life-Cycle-Phase Command . . . . .	3-46
8.1.1.10	Modifications to Copy Meta-Entity Command . . . . .	3-46
8.1.1.11	Modifications to Deactivate IRD Command . . . . .	3-47
8.1.1.12	Modifications to Restore IRD Schema Command . . . . .	3-47
8.1.1.13	Modifications to Activate IRD Command . . . . .	3-47
8.1.2	Modifications to Output IRD Schema Command . . . . .	3-47
8.2	Modifications to IRD Commands and Panel Trees . . . . .	3-48
8.2.1	Modifications to IRD Maintenance Commands . . . . .	3-53
8.2.1.1	Modifications to Add Entity Command . . . . .	3-53
8.2.1.2	Modifications to Modify Entity Command . . . . .	3-55
8.2.1.3	Modifications to Delete Entity Command . . . . .	3-57
8.2.1.4	Modifications to Add Relationship Command . . . . .	3-58
8.2.1.5	Modifications to Modify Relationship Command . . . . .	3-59
8.2.1.6	Modifications to Delete Relationship Command . . . . .	3-60
8.2.1.7	Modifications to Modify Entity Access-Name Command . . . . .	3-60
8.2.1.8	Modifications to Modify Entity Descriptive-Name Command . . . . .	3-61
8.2.1.9	Modifications to Modify Entity Life-Cycle-Phase Command . . . . .	3-61
8.2.1.10	Modifications to Copy Entity Command . . . . .	3-62
8.2.2	Modifications to IRD Output Functions . . . . .	3-64
8.2.2.1	Modifications to Output Impact-of-Change Command . . . . .	3-65
8.2.2.2	Modifications to Output Syntax Command . . . . .	3-66
8.3	Modifications to General Commands . . . . .	3-66
8.3.1	Modifications to Create IRD Command . . . . .	3-66
8.3.2	Modifications to Export IRD Command . . . . .	3-67
8.3.3	Modifications to Check IRD Schema Compatibility Command . . . . .	3-67
8.3.4	Modifications to Import IRD Command . . . . .	3-68
8.3.5	Modifications to Help Command . . . . .	3-68
8.3.6	Modifications to Enter Panel Dialogue Command . . . . .	3-68
8.4	Modifications to Panel Trees . . . . .	3-68
8.4.1	Modifications to Add Entity Panel Tree . . . . .	3-69
8.4.2	Modifications to Modify Entity Panel Tree . . . . .	3-69
8.4.3	Modifications to Modify Entity Life-Cycle-Phase Panel Tree . . . . .	3-69
8.4.4	Modifications to Copy Entity Panel Tree . . . . .	3-69



SECTION	PAGE
9 Modifications to the IRD Schema . . . . .	3-69
9.1 New Entity-Types . . . . .	3-70
9.2 New Relationship-Types . . . . .	3-70
9.3 New Attribute-Types . . . . .	3-71
9.4 New Attribute-Group-Types . . . . .	3-80
9.5 Changes to Entity-Types . . . . .	3-84
9.6 Entity-Type to Attribute-Type Meta-Relationships . . . . .	3-85
9.7 Entity-Type to Attribute-Group-Type Meta-Relationships . . . . .	3-87
9.8 Modification to Meta-Entity of Type IRDS-Reserved Names . . . . .	3-88
10 IRDS Security Panel Trees . . . . .	3-89
10.1 Add Security Panel Tree . . . . .	3-90
10.2 Modify Security Panel Tree . . . . .	3-91
10.3 Delete Security Panel Tree . . . . .	3-92
10.4 Add Access Key Panel Tree . . . . .	3-93
10.5 Delete Access Key Panel Tree . . . . .	3-94
11 Cross Reference Indices . . . . .	3-95
11.1 Alphabetic Index of Commands . . . . .	3-95
11.2 Alphabetic Index of Command-Clauses . . . . .	3-96
11.3 Error and Warning Condition Usage Index . . . . .	3-96
Appendix - Collected Syntax . . . . .	3-99
<b>Figures</b>	
Figure 1 - Architecture of the IRDS Security Facility . . . . .	3-2
Figure 2 - Processing of Commands by the IRDS Security Facility . . . . .	3-9
Figure 3 - IRD Schema Permission Structure . . . . .	3-13
Figure 4 - IRD Permission Structure . . . . .	3-14
<b>Tables</b>	
Table 1 - Suggested Review Sequence . . . . .	3-5



## **Chapter 3**

### **Module 3 - IRDS Security**

#### **1 Scope, Purpose, and Application**

##### **1.1 Scope**

The specification of the IRDS Security Facility defines the model and functionality of an access control facility which allows installations to restrict access to IRD and IRD Schema functionality and content.

##### **1.2 Purpose**

Module 3 of the standard IRDS defines an access control facility for the IRD and the IRD Schema. The access control facility can be regarded as being composed of the following components:

- (1) A set of permissions. These permissions define what actions individual IRDS-users can perform on both the IRD and the IRD Schema.
- (2) The descriptors (entity-types, relationship-types, attribute-group-types and attribute-types) which are used to define those permissions for individual IRDS-users.
- (3) The rules which govern the use of the commands and panel trees defined in Module 1 when access control rules are in effect.

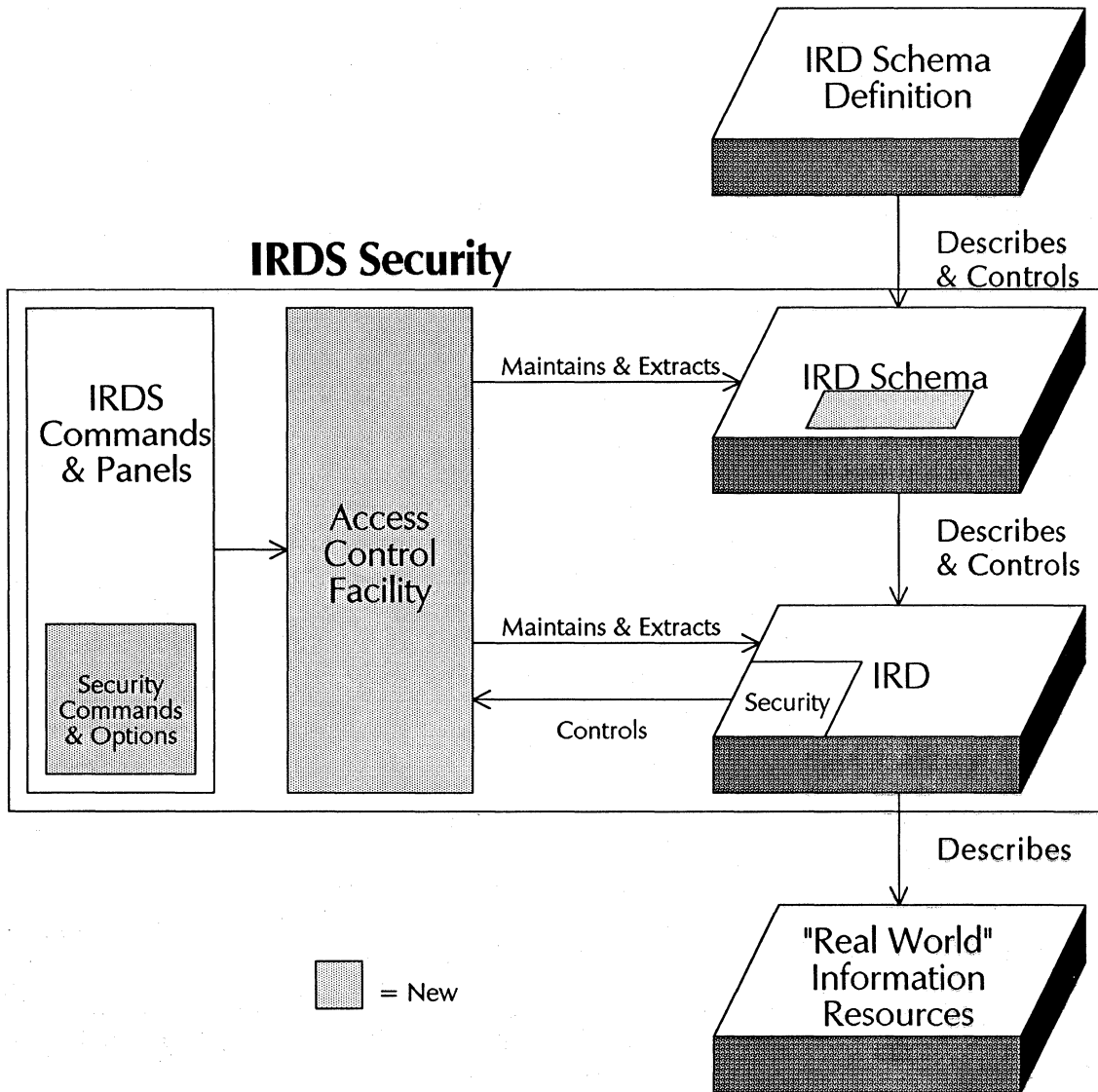
Additionally, some new commands, command-clauses and panel trees are also defined which are used for the administration of the permissions used by the access control facility.

##### **1.3 Architecture of the IRDS Security Facility**

The standard IRDS defines the structure of the IRD and IRD Schema, as well as the functionality to maintain, extract and report the contents of these. Figure 1 identifies the IRDS Security functionality which is relevant at each level.

As can be seen, Module 3 calls for the following changes to the Module 1 IRDS architecture:

- (1) There are changes in the content of the IRD Schema. New meta-entities and meta-relationships are defined; these descriptors define new attribute-types and associations of existing entity-types. These types define the structure of descriptors in the IRD partition named SECURITY.
- (2) There is a new access control facility. This facility determines whether actions resulting from either IRDS commands or panels are authorized or not. The specification of the access control facility consists of the Security Rules in this Module.



**Figure 1**  
**Architecture of the IRDS Security Facility**

The access control facility requires that "user profiles" exist. These "user profiles" shall exist in the SECURITY partition of the IRD.

The access control facility provides for two levels of access control: access control based on type and partition, and access control for individual entities. The former type of access control is called Global Security; the latter is called Entity-level Security. Global Security applies to both the IRD Schema and IRD. Entity-level Security applies only to the IRD, and is based on a scheme of access locks and keys. An entity which is "locked" can be accessed only through an IRD view which has a corresponding "key". Separate read and write locks and keys exist.

New commands and panel trees, as well as new options on some existing commands and panels have been added.

#### 1.4 Conformance

The Definition of a Conformant Implementation of Module 3 of a standard IRDS is given in the Definition of a Conformant Implementation of the IRDS (see Section 4 of Requirements for a Conformant Implementation).

#### 1.5 Organization

Module 3 of the standard IRDS consists of the following:

- (1) New Commands and Command-Clauses.
- (2) New Panel Trees which correspond to the new commands.
- (3) Modifications to the Syntax and Rules of Module 1.
- (4) Modifications to the IRD Schema.

The Command Language defined in this Module is an upward compatible extension to the Command Language defined in Module 1. Wherever appropriate, the syntax components defined in Module 1 are reused. Similarly, the new Panel Trees are also an upward compatible extension to those defined in Module 1.

The modifications to the syntax and rules of Module 1 take two forms:

- (1) Additional or modified rules for all commands and for specific commands.
- (2) Changes to the data areas in all existing panels as required by the new command options.

The modifications to the IRD Schema consist of:

- (1) One new entity-type (ACCESS-CONTROLLER)
- (2) A set of new relationship-types.
- (3) A set of new attribute-types.
- (4) A set of new attribute-group-types.
- (5) The meta-relationships between the new attribute-group-types and attribute-types.
- (6) The meta-relationships between new and existing entity-types and the new attribute-types and attribute-group-types.

Section 3 provides definitions of terms which are pertinent to this Module.

Section 4 provides an overview of the functionality provided by this Module. It includes a conceptual architecture for this Module which distinguishes between two layers of access control.

Section 5 defines the basic elements of the command language syntax. This section consists of new elements not defined in Module 1.

The new commands associated with this Module are defined in Section 6. The format is the same as found in Module 1 for command specifications.

The command-clauses which are unique to this Module are defined in Section 7.

The modifications to the Command Language and Panel Trees of Module 1 are given in Section 8.

The modifications to the IRD Schema are described in Section 9. The modifications are given in the form of IRD Schema commands.

Section 10 defines the panel trees which correspond to the new commands described in Section 6.

Section 11 provides three indices:

- (1) An alphabetical index of all commands defined in this Module.
- (2) An alphabetical index of all command-clauses defined in this Module.
- (3) A cross-reference index of the error and warning conditions. This index lists each error and warning condition by its identifier, and identifies each section or subsection where the condition occurs.

Table 1 presents the suggested order in which this chapter should be reviewed. It is assumed that the reviewer is familiar with Module 1 of the standard IRDS.

Note that Sections 5 through 8 define commands and syntax. The introductory material in Section 8 provides the formal definition of permissions and how these permissions relate one to another. This material should be read before attempting to read the description of any command or command modification. Sections 5 and 7 define common syntax elements which are referenced in Sections 6 and 8. Accordingly, as readers follow the command syntax to the lowest level of detail, they will cover Sections 5 and 7.

## 1.6 Notation

This Module uses the same notational conventions as identified in Module 1 of the standard IRDS, except as noted below.

The Commands and Command-Clauses defined in this Module have cross-references to syntax defined both in this Module and in other Modules. Whenever a cross reference number is given as a sequence of numbers alone, the reference is to a section or subsection within this Module. References to another Module begin with  $M_n$ , where  $n$  denotes the number of the other Module of the standard IRDS.

Thus:

"6.2" denotes a reference to Subsection 6.2 of this Module;

**Table 1**  
**Suggested Review Sequence**

ORDER	SECTION	TITLE	ACTION
1	1	SCOPE, PURPOSE AND APPLICATION	READ
2	2	REFERENCES	SKIM
3	4	MODULE OVERVIEW	READ
4	8	MODIFICATIONS TO MODULE 1 COMMANDS AND PANEL TREES	READ
5	6	IRDS SECURITY COMMANDS	READ
6	7	IRDS SECURITY COMMAND-CLAUSES	AS REQUIRED
7	5	BASIC LANGUAGE ELEMENTS	AS REQUIRED
8	10	IRDS SECURITY PANEL TREES	READ
9	9	MODIFICATIONS TO THE IRD SCHEMA	SKIM
10	3	DEFINITIONS	AS REQUIRED
11	11	CROSS REFERENCE INDICES	AS REQUIRED

whereas

"M1.4.3" is a reference to Subsection 4.3 of Module 1.

Error and warning conditions have the following format:

- (1) One alphabetic character which defines the message level:
  - (a) E identifies an error.
  - (b) W identifies a warning.
- (2) Two digits identify the Module of the standard IRDS where the message originates.
- (3) Three digits identify the message number within the Module of the standard IRDS and type of condition. (Thus error and warning conditions may have the same last three digits, i.e., E01001 and W01001 are two different conditions.)

## 2 References

None.

## 3 Definitions

**access controller.** (1) In an IRD with a security partition, a pair of locks, one for read access, the other for write access. (2) In an IRD with a security partition, a security entity of the type ACCESS-CONTROLLER.

**access key.** In an IRD with a security partition, an authorization to perform a set of operations on an entity secured by a lock.

**default IRD-schema-view.** The IRD-schema-view which automatically becomes the effective IRD-schema-view when an IRDS-user is recognized by the IRDS as ready to submit IRDS transactions.

**default IRD-view.** The IRD-view which automatically becomes the effective IRD-view when an IRDS-user is recognized by the IRDS as ready to submit IRDS transactions.

**effective IRD-schema-view.** An IRD-schema-view which identifies the IRD Schema life-cycle-phase in which entities to be maintained exist. The effective IRDS-user can have only one effective IRD-schema-view at any one time. A user can change the effective IRD-schema-view with the set-session-defaults command.

**effective IRD-view.** An IRD-view which identifies the IRD-partition in which entities to be maintained exist. The effective IRDS-user can have only one effective IRD-view at any one time. A user can change the effective IRD-view with the set-session-defaults command.

**effective IRDS-user.** An IRDS-user entity which is associated by the IRDS with one or more IRDS transactions.

**entity-level security.** The facilities of the IRDS which control access to entities on an individual entity basis.

**global security.** The access-control facilities which are common to IRD Schema and IRD transactions. In global security, certain permissions are defined for an IRDS-user. Other permissions are for any given IRDS-user are defined on related IRD-views and IRD-schema-views. Those defined in an IRD-view authorize transactions based on the type of entity, the type of transaction, and the partition in which the entity exists. Similarly, permissions defined in an IRD-schema-view authorize transactions based on the type of meta-entity, the type of transactions, and the IRD Schema life-cycle-phase in which the meta-entity exists.

**IRD life-cycle-phase.** An IRD partition which represents a life-cycle-phase.

**IRD partition.** A logical subset of IRD entities. Each IRD entity exists in one and only one IRD partition. Every IRD life-cycle-phase is an IRD partition. There can exist other IRD partitions for special purposes. One such IRD partition is named SECURITY. All security



entities are contained in this IRD partition. Each IRD partition is defined in the IRD Schema by a meta-entity of type IRD-PARTITION.

**IRD Schema life-cycle-phase.** One of the three life cycle-phases which are defined for meta-entities. These life cycle phases are called UNCONTROLLED, CONTROLLED, and ARCHIVED.

**IRD-schema-view.** A security entity of type IRD-SCHEMA-VIEW. An IRD-schema-view defines a set of permissions which apply within a given IRD Schema life-cycle-phase. In order to be usable, an IRD-schema-view must be related to at least one IRDS-user. An IRDS-user can have multiple IRD-schema-views.

**IRD-view.** A security entity of type IRD-VIEW. An IRD-view defines a set of permissions which apply within a given IRD partition. In order to be usable, an IRD-view must be related to at least one IRDS-user. An IRDS-user can have multiple IRD-views.

**IRDS-user.** A security entity of type IRDS-USER. An entity of this type represents an individual or group of individuals who are authorized to use the IRDS in some capacity. Global permissions for the individual(s) represented by an IRDS-user are defined with attributes of this entity.

**key.** See access-key.

**life-cycle.** A conceptual framework which is used to trace the evolution of objects of certain classes over time. A life-cycle is divided into phases. An object is identified as existing in a particular life-cycle-phase. An object shall exist in only one life-cycle-phase at given instance in time.

**life-cycle-phase.** A major division of a life-cycle.

**lock.** A mechanism to restrict authorization for a set of operations against a specific entity to a particular entity of the IRD.

**permission.** An authorization to perform a certain set of operations.

**permission structure.** A partial ordering of permissions. Given two permissions *A* and *B*, the ordering is defined as follows:

- (1) If  $A > B$ , then if *A* is granted, *B* is also granted.
- (2) If *A* and *B* are not comparable (i.e., *A* is not greater than *B* and *B* is not greater than *A*), then the permissions are independent.

**read key.** An access-key which authorizes the ability to select an entity with the corresponding read-lock and test and/or display all of that entity's attributes.

**read lock.** A lock which denies authorization to select an entity and test and/or display its attributes.

**secured entity.** In an IRD with a security partition, an entity which is not a security entity and which is related to at least one access controller.

**security entity.** An entity which exists in the IRD partition named SECURITY and which is required to support the IRDS security functionality.

**security maintenance set.** For any given IRDS-user *U*, this is the set of all access-controllers which *U* is authorized to assign or deassign from any entity.

**security partition.** In an IRD, the IRD partition containing all security entities.

**visibility.** The ability of a user to select and display the following: An entity and its attributes; a relationship and its attributes; a meta-entity and its meta-attributes, and a meta-relationship and its attributes.

- (1) Visibility is a precondition for maintenance authorization. Subsection 8.1 provides the formal definitions for visibility of meta-entities and meta-relationships. Subsection 8.2 provides the formal definitions for entities and relationships.
- (2) The visibility of meta-entities for a given user is through an IRD-schema-view which is related to the corresponding IRDS-user entity. Visibility of meta-relationships requires one or two IRD-schema-views, depending on where the component entities reside.
- (3) Similarly, visibility of entities is through an IRD-view related to the corresponding IRDS-user entity, and relationships can require one or two IRD-views to be visible to the user. The visibility of an entity or relationship depends both on global and entity-level security considerations.

**write key.** An explicit authorization to modify or delete entities which have been assigned the corresponding write-lock. A write-key also grants the same privileges as a read-key.

**write-lock.** A lock which denies authorization to modify or delete an entity.

## 4 Module Overview

This Module specifies facilities for controlling access to the IRD Schema and IRD.

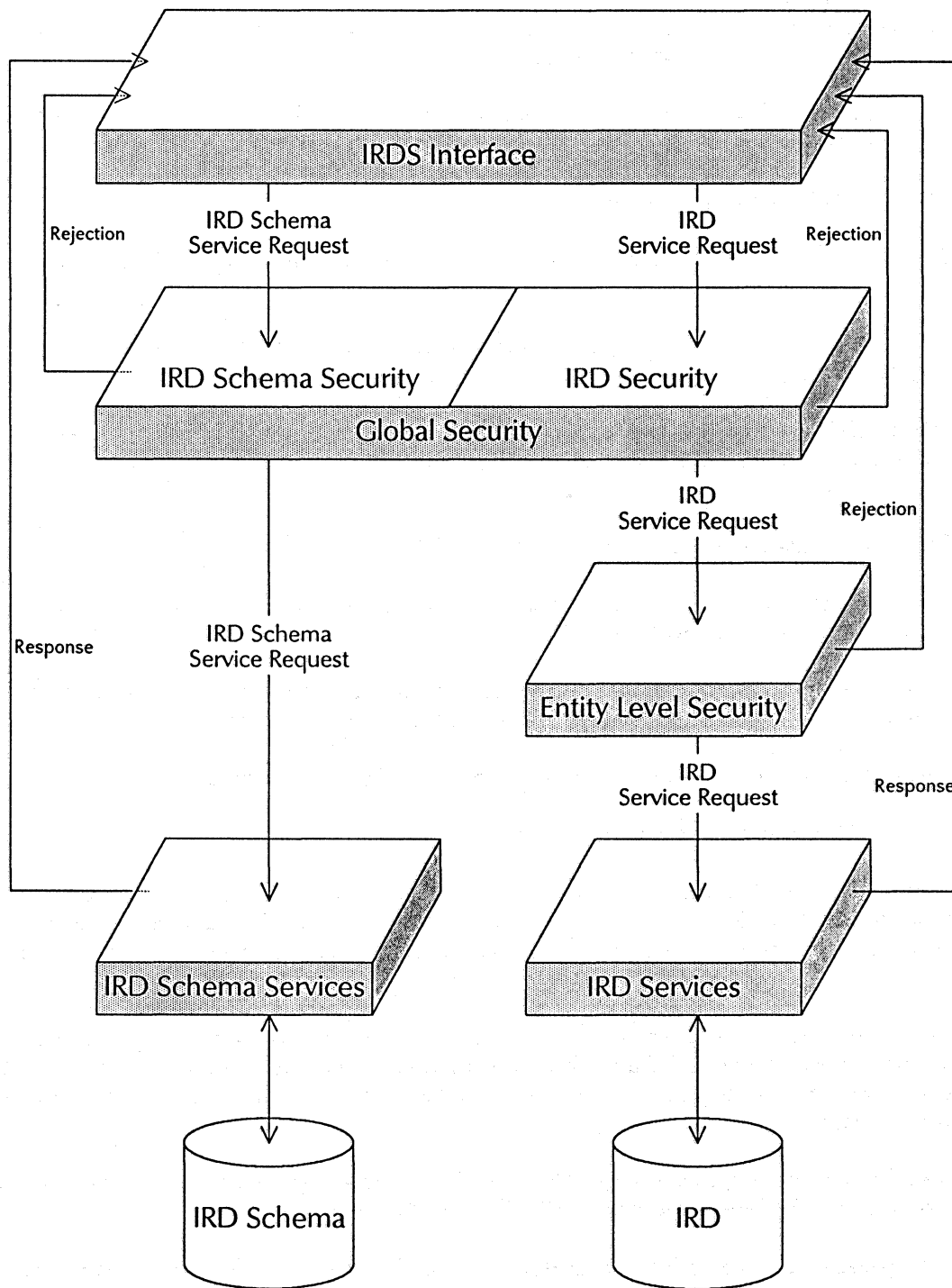
### 4.1 Description of Security Facilities

This Module specifies IRDS security facilities by means of two layers of security. The first layer provides the Global Security facility. The second layer provides the Entity-Level Security facility.

The Global Security facility applies to both the IRD Schema and IRD and the commands that operate on them. The Entity-Level Security facility applies only to the IRD and the commands that operate on it.

A command operating on the IRD which is issued by an IRDS user shall first be checked by the Global Security facility. If the command passes this check it shall then be passed to the Entity-Level Security facility for checking only after it has passed through the checking performed by Global Security. Figure 2 illustrates the interaction between these two layers for processing of IRD and IRD Schema commands.

The two layers operate differently.



**Figure 2**  
**Processing of Commands by the IRDS Security Facility**

- (1) The Global Security facility constitutes a closed security system. This means that access to IRD Schema and/or IRD descriptors shall be granted on a meta-entity-type or entity-type basis, respectively, only if explicitly authorized.
- (2) The Entity-Level Security facility constitutes an open security system. This means that access to IRD descriptors shall be granted unless explicitly denied. This facility provides the means of excluding specific entities from IRD-views.

Both layers use the IRD view concept. IRD-schema-views are defined for the IRD Schema, and IRD-views are defined for the IRD. IRD-views and IRD-schema-views have similar structure.

#### **4.1.1 The Global Security Facility**

Each IRD Schema view is associated with an IRD Schema life-cycle-phase. IRD Schema views define:

- (1) A set of permissions for the actions that can be taken by an IRDS-user on meta-entities of specified types within an IRD Schema life-cycle-phase.
- (2) Meta-relationship-types, instances of which are excluded from the IRD-schema-view whenever at least one of the participating meta-entities is in the associated IRD Schema life-cycle-phase.

Each IRD-view is associated with an IRD partition. IRD-views are used to define:

- (1) A set of permissions for the actions that can be taken by an IRDS-user on entities of specified types within an IRD partition.
- (2) Relationship-types, instances of which are excluded from the IRD-view whenever at least one of the participating entities is in the associated IRD partition.

Other permissions are declared as attributes of entities of type IRDS-user.

#### **4.1.2 The Entity-Level Security Facility**

The Entity-Level Security Facility allows access to individual IRD entities to be restricted. An entity shall be secured when an access-controller is assigned to it. The same access-controller can be assigned to multiple entities, and multiple access-controllers can be assigned to a secured entity.

An access-controller shall be assigned a pair of locks: a read-lock, and a write-lock. Securing an entity may be visualized as building a fence with two locked gates around the entity. One lock applies to operations that require the entity to be read; the other applies to actions that write to the entity.

A gate can be opened by an access-key. There shall be a read-key which "opens" the read-lock, and a write-key which "opens" both the write-lock and the read-lock.

In order to perform a command which updates a secured entity, the IRD view in effect shall have a write-key assigned to it which corresponds to one of the access-controllers which is assigned to the entity.

In order to perform a command which reads a secured entity, then either:

- (1) The user shall specify an IRD-view in a using-IRD-views-clause which has either a read-key or write-key for one of the access-controllers assigned to the entity; or
- (2) The IRD-view in effect shall be assigned a read-key or write-key for one of the access-controllers assigned to the entity.

If these conditions are not met, then the secured entity shall be treated as if it did not exist in the specified IRD-view.

Any entity, with exception of the security entities themselves, can be secured. Security entities shall be those which exist in the security partition.

#### 4.2 Security Descriptors

All descriptors used by the IRDS Security Facility shall be stored and maintained in the IRD. Any entity which is used by the security facility is called a security entity. An entity of one of the following types is a security entity:

IRDS-USER  
 IRD-VIEW  
 IRD-SCHEMA-VIEW  
 ACCESS-CONTROLLER

Such an entity shall exist in the IRD partition named SECURITY. The following relationship-types support global security:

IRDS-USER-HAS-IRD-SCHEMA-VIEW  
 IRDS-USER-HAS-IRD-VIEW

This Module defines new attribute-types that shall be associated with the entity-types IRDS-USER, IRD-VIEW, and IRD-SCHEMA-VIEW, which support the global security facilities. The ACCESS-CONTROLLER entity-type is required for the entity-level security facilities.

The following new attribute-types associated with the entity-type IRDS-USER shall be used to specify permissions for an IRDS-user:

IRD-ADMINISTRATOR-PERMISSION  
 COMMAND-LANGUAGE-PERMISSION  
 PANEL-PERMISSION  
 IRD-RENAME-PERMISSION  
 IRD-SCHEMA-ADMINISTRATOR-PERMISSION  
 IRD-SCHEMA-RENAME-PERMISSION

The following new attribute-types associated with the entity-type IRD-VIEW shall be used to specify permissions for an IRDS-user using an IRD-view:

IRD-ADMINISTRATOR-PERMISSION  
 ACCESS-KEYS-PERMISSION

Additionally, the following new attribute-types are components on a new attribute-group-type which is used to specify the actions permitted on entities of a given type:

READ-PERMISSION  
ADD-PERMISSION  
MODIFY-PERMISSION  
DELETE-PERMISSION  
MODIFY-PHASE-PERMISSION  
ADD-AND-SECURE-PERMISSION  
SECURE-ENTITY-PERMISSION

The IRD-VIEW entity-type is also associated with attribute-types used to control access to entities secured on an individual entity basis.

The following new attribute-type associated with the entity-type IRD-SCHEMA-VIEW is used to specify a permission for an IRDS-user using an IRD-schema-view:

IRD-SCHEMA-ADMINISTRATOR-VIEW

Additionally, the following attribute-types are also components on a new attribute-group-type which is used to specify the actions permitted on meta-entities of a given type:

READ-PERMISSION  
ADD-PERMISSION  
MODIFY-PERMISSION  
DELETE-PERMISSION

Note that the attribute-types IRD-ADMINISTRATOR-PERMISSION and IRD-SCHEMA-ADMINISTRATOR-PERMISSION shall be associated with both the entity-type IRDS-USER and the entity-types IRD-VIEW and IRD-SCHEMA-VIEW, respectively. In this manner these permissions can be assigned to an IRDS-user for all IRD-views or IRD-schema-views belonging to that user, or only to specific views.

These permissions are ordered in a partial ordering; in this ordering a permission which is superior shall imply all inferior permissions. This partial ordering is shown in Figures 3 and 4.

Additionally, relationships belonging to the relationship-class-type SECURED-BY support entity-level security. There shall exist one relationship-type in this class for each entity-type which is not a security entity-type. The names of these relationship-types all have the form

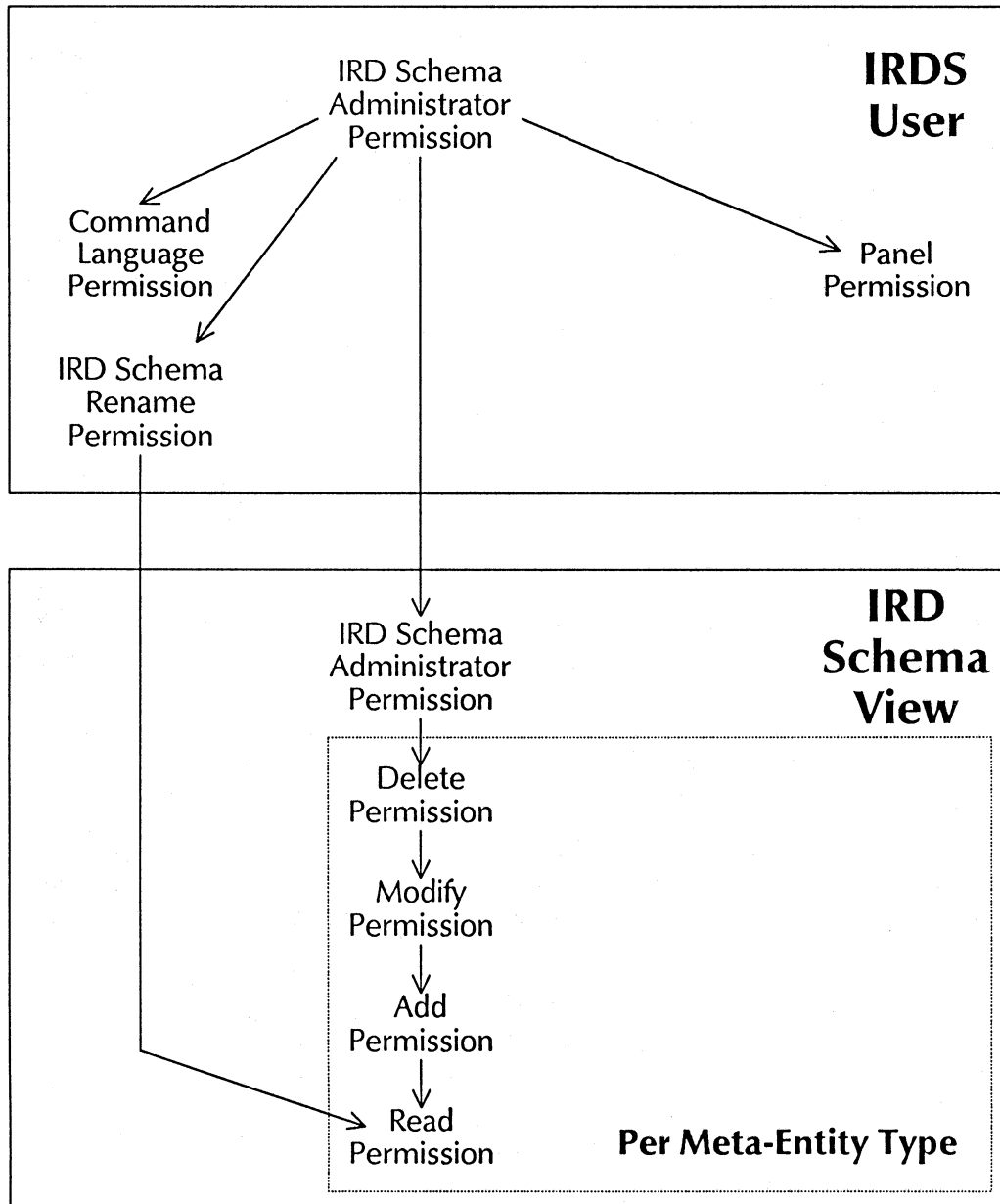
X-SECURED-BY-ACCESS-CONTROLLER

where *X* is the name of the entity-type. These relationship-types can be established by an installation whenever it wishes to make entity-level security available for entities of type *X*.

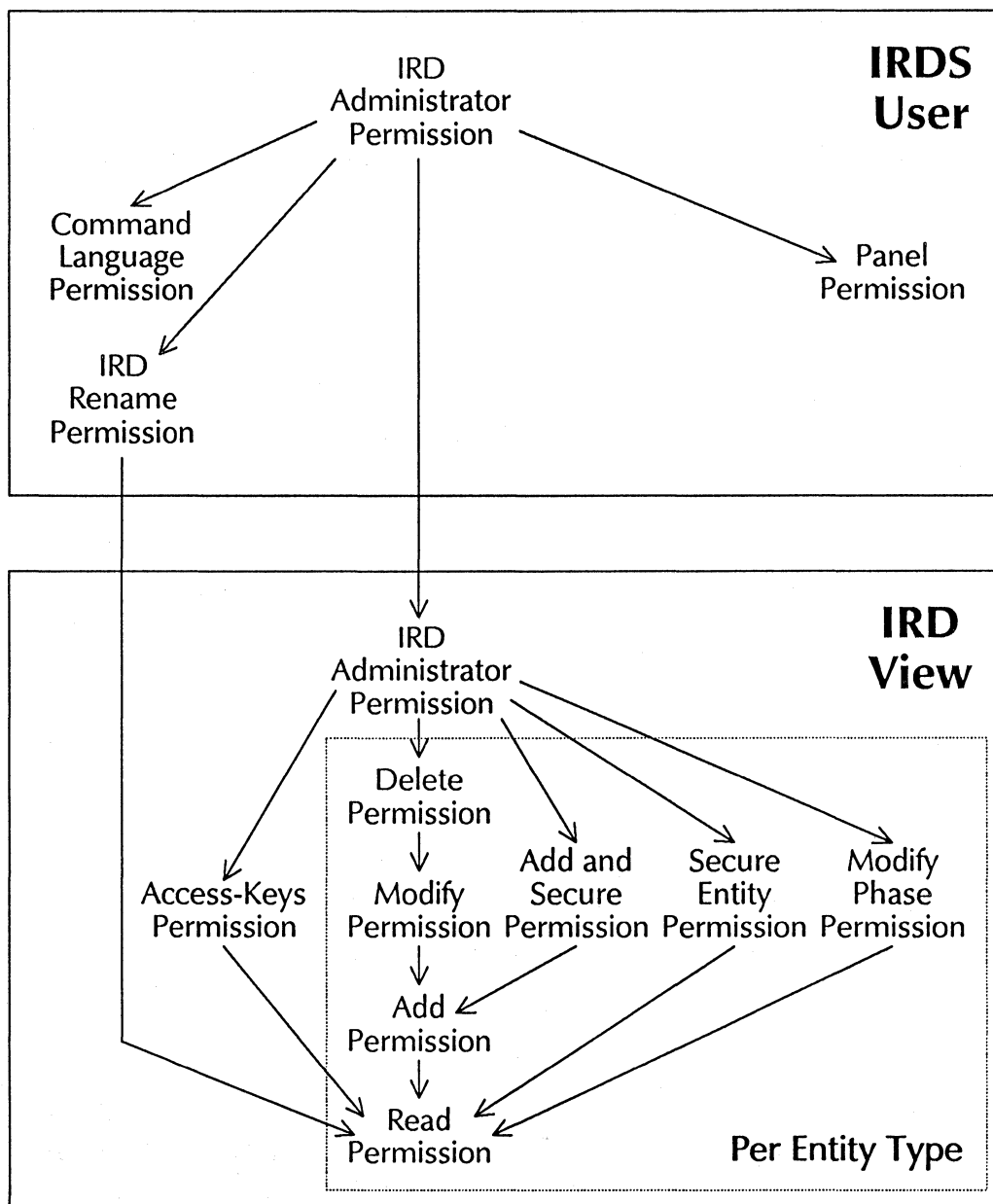
Read access to all security descriptors shall be controlled by the global security facilities. The IRD maintenance commands defined in Module 1 shall be used to maintain global security descriptors (IRDS-users, IRD-views, IRD-schema-views, and the corresponding relationships). The global security facility controls write access to these security descriptors.

The entity-level security-descriptors, (entities of type ACCESS-CONTROLLER and SECURED-BY relationships) shall be maintained only by the commands provided by this Module. These commands shall only be executed if the effective IRDS user has the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES.

The assignment of an access-controller to an entity of type *X* is implemented by a relationship of type X-SECURED-BY-ACCESS-CONTROLLER. Thus:



**Figure 3**  
**IRD Schema Permission Structure**



**Figure 4**  
**IRD Permission Structure**



- (1) The assignment of an access-controller *c* to an entity *x* of type *X* creates the relationship:

*x* SECURED-BY *c*.

- (2) The deletion of security for entity *x* shall be accomplished by deleting such relationships.

Locks and access-keys shall be implemented as follows:

Each entity of type ACCESS-CONTROLLER has two attributes. One is of type WRITE-LOCK, the other is of type READ-LOCK. They shall be 10 digit numeric strings. Each lock attribute has a unique value. The values assigned to these attributes shall be established by the IRDS. Values shall be assigned serially.

The implementation of access-keys shall be based on two repeating attribute-types associated with the IRD-VIEW entity-type. The first attribute-type is READ-KEY, and the second is WRITE-KEY. Attributes of these types shall be entity-level security descriptors and shall not be visible to the normal IRD commands. Thus they shall not be maintained by normal IRD maintenance commands and/or panels.

When an IRD-view is assigned an access-key for an access-controller, the value of the corresponding lock attribute shall be assigned to the access-key attribute.

## 5 Basic Language Elements

### 5.1 Function

To identify those additional terminal and nonterminal symbols which are provided with this Module.

### 5.2 Format

access-controller-name ::= assigned-access-name M1.4.4  
 IRD-view-name ::= assigned-access-name M1.4.4

/\* These words are additional irds-words. The presentation format is the same as for irds-words. \*/

access-key-word ::= ACCESS-KEY  
 access-keys-word ::= ACCESS-KEYS  
 assign-word ::= ASSIGN  
 controller-word ::= ACCESS-CONTROLLER  
 controllers-word ::= CONTROLLER  
 controller-word ::= controller-word  
 ACCESS-CONTROLLERS  
 CONTROLLERS  
 read-word ::= READ  
 security-word ::= SECURITY  
 write-word ::= WRITE

```
/* Additional irds-reserved-words */
   irds-module-3-reserved-words := controller-word
```

### 5.3 Syntax Rules

- (1) The singular form of a word is provided to enhance syntax readability. In such cases, the singular and plural forms shall be equivalent.

5.4 General Rules None.

5.5 Error and Warning Conditions None.

## 6 IRDS Security Commands

The commands described in this and subordinate subsections are specialized IRD maintenance commands. Accordingly, all rules and error and warning conditions which are described in Section 8 of this Module apply to this section and its subordinate subsections. Also, all rules and error and warning conditions described in Section 5 of Module 1, which are not overridden by this Module shall also apply to this and subordinate subsections.

The use of these commands shall be governed by Security Rules. These rules are written presuming knowledge of the structure of IRD permissions. The description of these permissions is given in Subsection 8.2 of this Module.

**Function** To identify those commands which manage the security of individual entities.

### Format

/\* None. This breakdown is explanatory only. \*/

```
irds-security-command ::=
    add-security-command           6.1
  | modify-security-command       6.2
  | delete-security-command       6.3
  | add-access-key                6.4
  | delete-access-key             6.5
```

The following notation is used in defining Rules in this and subordinate Subsections:

*U* denotes the effective IRDS-user.  
*V* denotes the effective IRD-view.  
*P* denotes the IRD-partition associated with *V*.

**Syntax Rules** None.

### General Rules

- (1) *V* shall have the attribute IRD-PARTITION-NAME equal to SECURITY (i.e., *P* = SECURITY).

## Security Rules

- (1) If  $U$  has the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES, then  $U$  shall be authorized use of these commands. Additional conditions which apply to specific commands are identified in the Security Rules for those commands.
- (2) The security maintenance set for  $U$  refers to the set of access-controllers which the user can assign to or deassign from any entity. This set of access-controllers  $S$  is defined to be as follows:
  - (a) If  $U$  has the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES, then  $S$  shall equal the set of all access-controllers in the IRD.
  - (b) If  $V$  authorizes read-permission for entities of type ACCESS-CONTROLLER,  $S$  shall be defined as follows:
 

Let  $V_i$ ,  $1 \leq i \leq n$ , denote the IRD-views which are related to  $U$  and which are not associated with the IRD partition named SECURITY. Let  $C_{ij}$ ,  $1 \leq j \leq m_i$  denote the access-controllers for which the view  $V_i$  has an access-key. Then  $S$  is the union of all  $C_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq m_i$ .
  - (c) Otherwise,  $S$  shall be the null set.

For all other rules in this and subordinate Subsections, let  $S$  denote the security maintenance set for  $U$ .

## Error and Warning Conditions

- (1) Error E03017: Effective IRD-view must be associated with SECURITY partition. See General Rule (1).

## 6.1 Add Security Command

**Function** To assign an access-controller to one or more entities.

### Format

```

add-security-command ::=
    add-word                M1.4.3
    security-word           5
    [ to-word ]            M1.4.3
    secured-entities-option
    controller-list-clause  7.5
    ;

secured-entities-option ::=
    access-name-list
    | entities-in-clause   7.1

access-name-list ::=
    access-name            M1.4.4
    [ , access-name-list ]
  
```

**Syntax Rules** None.

**General Rules**

- (1) Each access-name specified within access-name-list shall identify an entity existing in the IRD.
- (2) If any entity within the SECURITY partition is either specified in or selected via secured-entities-option, an error condition shall exist for that entity.
- (3) If an access-controller *A* is assigned to an entity *E* which is specified in or selected via secured-entities-option, a warning shall be issued.
- (4) Assume *E* is an entity specified or selected in secured-entities-option, and *E* is of type *T*. Then the following meta-entities shall exist in the CONTROLLED IRD Schema life-cycle-phase:
  - (a) An entity-type with assigned-access-name equal to the assigned-access-name of *T*.
  - (b) An entity-type with assigned-access-name equal to ACCESS-CONTROLLER.
  - (c) A relationship-class-type with assigned-access-name equal to SECURED-BY.
  - (d) A relationship-type which is a member of the relationship-class-type SECURED-BY with *T* as the first participating entity-type and ACCESS-CONTROLLER is the second participating entity-type.

Each of the these meta-entities shall be as defined in Section 9 of this Module.

**Security Rules**

For the following Security Rules:

Let *E<sub>i</sub>* denote an entity to be secured.  
Let *T<sub>i</sub>* denote the type of *E<sub>i</sub>*.  
Let *P<sub>i</sub>* denote the IRD-partition in which *E<sub>i</sub>* exists.

- (1) If *U* does not have the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES, then for each *P<sub>i</sub>*, there shall be an IRD-view *V<sub>i</sub>* such that the following are satisfied:
  - (a) *V<sub>i</sub>* is related to *U*.
  - (b) *V<sub>i</sub>* is associated with *P<sub>i</sub>*.
  - (c) One of the following is satisfied:
    - (i) *V<sub>i</sub>* has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
    - (ii) *V<sub>i</sub>* has an IRD-PERMISSIONS attribute-group with the ENTITY-TYPE-NAME attribute equal to ALL or the assigned-access-name of *T<sub>i</sub>* and the SECURE-ENTITY-PERMISSION attribute equal to YES.

If the above conditions are met, it is said that *U* is authorized SECURE-ENTITY-PERMISSION for entities of type *T<sub>i</sub>* in *P<sub>i</sub>*.

- (2) If  $U$  is authorized IRD-ADMINISTRATOR-PERMISSION, then  $U$  shall be authorized SECURE-ENTITY-PERMISSION to all entities in the IRD except as excluded by General Rule (2) above.
- (3) Each  $Ei$  shall be visible to  $U$ . (See Subsection 8.2 for the definition of visibility.) In particular:
  - (a) If  $Ei$  is not secured, no additional requirements have to be satisfied.
  - (b) If  $Ei$  is to be assigned access-controllers  $Ck$ ,  $1 \leq k \leq p$ , then one of the following shall be satisfied:
    - (i)  $U$  or  $Vi$  has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
    - (ii)  $Vi$  has the read or write key for some access-controller  $Ck$ ,  $1 \leq k \leq p$ .
- (4) Each access-controller specified in controller-list-clause shall be a member of  $S$ .

#### **Actions**

- (1) Syntax validation shall be performed. All appropriate error and warning messages shall be issued.
- (2) If any errors were encountered, the command shall be terminated.
- (3) Entities shall be selected. For each selected entity, each specified access-controller shall be assigned, unless it already is assigned. If an error occurs in processing any given entity, then no access-controller shall be assigned to that entity. All appropriate error and warning messages shall be issued for that entity. The command shall proceed to the next specified or selected entity.
- (4) End of command processing shall be confirmed.

#### **Error and Warning Conditions**

- (1) Error E01009: Entity does not exist or is not visible. See General Rule (1).
- (2) Error E03007: Cannot secure a security entity. See General Rule (2).
- (3) Warning W03004: Entity already assigned specified access-controller. See General Rule (3).
- (4) Error E03008: Entity-level security cannot be assigned to specified entity. See General Rule (4).
- (5) Error E03011: User is not authorized to perform specified action(s) by effective IRD-view. See Security Rule (1) and Security Rule (3)(b)(ii).
- (6) Error E03004: User not authorized to perform specified action(s). (The user is not authorized SECURE-ENTITY-PERMISSION in any IRD partition.) See Security Rules (1) and (2).
- (7) Error E03025: User cannot assign/deassign specified access-controller. See Security Rule (4).

## 6.2 Modify Security Command

**Function** For one or more entities, to deassign a specified access-controller and assign another access-controller in its place.

### Format

```

modify-security-command ::=
    modify-word                M1.4.3
    security-word              5
    [ of-word ]                M1.4.3
    secured-entities-option
    from-controller-clause     7.2
    to-controller-clause      7.3
    ;

secured-entities-option ::=
    access-name-list
    | entities-in-clause      7.1

access-name-list ::=
    access-name                M1.4.4
    [ , access-name-list ]

```

**Syntax Rules** None.

### General Rules

- (1) Each access-name specified within access-name-list shall identify an entity existing in the IRD.
- (2) This command shall not operate on any entity within the IRD-partition named SECURITY. If any entity within the SECURITY partition is either specified in or selected via secured-entities-option, an error condition shall exist for that entity.
- (3) For any specified or selected entity *E*, both of the following shall be satisfied:
  - (a) The access-controller specified in from-controller-clause shall be assigned to *E*.
  - (b) The access-controller specified in to-controller-clause shall not be assigned to *E*.
- (4) Assume *E* is an entity specified or selected in secured-entities-option, and *E* is of type *T*. Then the following meta-entities shall exist in the CONTROLLED IRD Schema life-cycle-phase:
  - (a) An entity-type with assigned-access-name equal to the assigned-access-name of *T*.
  - (b) An entity-type with assigned-access-name equal to ACCESS-CONTROLLER.
  - (c) A relationship-class-type with assigned-access-name equal to SECURED-BY.
  - (d) A relationship-type which is a member of the relationship-class-type SECURED-BY with *T* as the first participating entity-type and ACCESS-CONTROLLER is the second participating entity-type.

Each of the these meta-entities shall be as defined in Section 9 of this Module.

### Security Rules

For the following Security Rules:

Let  $E_i$  denote an entity to be secured.  
 Let  $T_i$  denote the type of  $E_i$ .  
 Let  $P_i$  denote the IRD-partition in which  $E_i$  exists.

- (1) If  $U$  does not have the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES, then for each  $P_i$ , there shall be an IRD-view  $V_i$  such that the following are satisfied:
  - (a)  $V_i$  is related to  $U$ .
  - (b)  $V_i$  is associated with  $P_i$ .
  - (c) One of the following is satisfied:
    - (i)  $V_i$  has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
    - (ii)  $V_i$  has an IRD-PERMISSIONS attribute-group with the ENTITY-TYPE-NAME attribute equal to ALL or the assigned-access-name of  $T_i$  and the SECURE-ENTITY-PERMISSION attribute equal to YES.

If the above conditions are met, it is said that  $U$  is authorized SECURE-ENTITY-PERMISSION for entities of type  $T_i$  in  $P_i$ .

- (2) If  $U$  is authorized IRD-ADMINISTRATOR-PERMISSION, then  $U$  shall be authorized SECURE-ENTITY-PERMISSION to all entities except as excluded by General Rule (2) above.
- (3) Each  $E_i$  shall be visible to  $U$ . (See Subsection 8.2 for the definition of entity visibility.) In particular:
  - (a) If  $E_i$  is not secured, no additional requirements need be satisfied.
  - (b) If  $E_i$  is to be assigned access-controllers  $C_k$ , then one of the following shall be satisfied:
    - (i)  $U$  or  $V_i$  has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
    - (ii)  $V_i$  has the read or write key for some access-controller  $C_k$ .
- (4) The controller specified in from-controller-clause shall be a member of  $S$ .
- (5) The controller specified in to-controller-clause shall be a member of  $S$ .

### Actions

- (1) Syntax validation shall be performed. All appropriate error and warning messages shall be issued.
- (2) If any errors were encountered, the command shall be terminated.

- (3) Entities shall be selected. For each selected or specified entity, access-controllers shall be reassigned, unless an error condition is encountered for that specific entity. All appropriate error and warning messages shall be issued for that entity. The command shall proceed to the next specified or selected entity.
- (4) End of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01009: Entity does not exist or is not visible. See General Rule (1).
- (2) Error E03007: Cannot secure a security entity. See General Rule (2).
- (3) Error E03018: From-controller not assigned to entity. See General Rule (3)(a).
- (4) Error E03019: To-controller already assigned to entity. See General Rule (3)(b).
- (5) Error E03008: Entity-level security cannot be assigned to specified entity. See General Rule (4).
- (6) Error E03011: User is not authorized to perform specified action(s) by effective IRD-view. See Security Rule (1) and Security Rule (3)(b)(ii).
- (7) Error E03004: User not authorized to perform specified action(s). (The user is not authorized SECURE-ENTITY-PERMISSION in any IRD partition.) See Security Rules (1) and (2).
- (8) Error E03025: User cannot assign/deassign specified access-controller. See Security Rules (4) and (5).

**6.3 Delete Security Command**

**Function** To remove either specific or all entity-level security constraints from one or more entities.

**Format**

```

delete-security-command ::=
    delete-word           M1.4.3
    security-word         5
    [ of-word ]          M1.4.3
    secured-entities-option
    related-controllers-clause 7.4
    ;

secured-entities-option ::=
    access-name-list
    | entities-in-clause 7.1

access-name-list ::=
    access-name           M1.4.4
    [ , access-name-list ]
    
```



**Syntax Rules** None.

### General Rules

- (1) Each access-name specified within access-name-list shall identify an entity existing in the IRD.
- (2) This command shall not operate on any entity within the IRD-partition named SECURITY. If any entity within SECURITY partition is either specified in or selected via secured-entities-option, an error condition shall exist for that entity.
- (3) If all-word is specified in related-controllers-clause, then for specified entity  $E$ , all access-controllers assigned to  $E$  in  $S$  shall be deassigned.
  - (a) If the entity is still secured by an access-controller which is not in  $S$ , a warning shall be issued.
  - (b) If  $E$  does not have any controllers assigned, then no action shall be taken with respect to  $E$  and a warning shall be issued.

In either case, processing shall continue with the next entity.

- (4) If related-controllers-clause specifies access-controllers  $C1 \dots Cn$ , then for each specified entity  $E$ :
  - (a) Each  $Ci$ ,  $1 \leq i \leq n$ , which is assigned to  $E$  shall be deassigned.
  - (b) If any  $Cj$ ,  $1 \leq j \leq n$ , is not assigned to  $E$ , then no action shall be taken with respect to  $Cj$  and  $E$  and a warning shall be issued.
- (5) Assume  $E$  is an entity specified or selected in secured-entities-option, and  $E$  is of type  $T$ . Then the following meta-entities shall exist in the CONTROLLED IRD Schema life-cycle-phase:
  - (a) An entity-type with assigned-access-name equal to the assigned-access-name of  $T$ .
  - (b) An entity-type with assigned-access-name equal to ACCESS-CONTROLLER.
  - (c) A relationship-class-type with assigned-access-name equal to SECURED-BY.
  - (d) A relationship-type which is a member of the relationship-class-type SECURED-BY with  $T$  as the first participating entity-type and ACCESS-CONTROLLER is the second participating entity-type.

Each of the these meta-entities shall be as defined in Section 9 of this Module.

### Security Rules

For the following Security Rules:

Let  $Ei$  denote an entity to be secured.  
 Let  $Ti$  denote the type of  $Ei$ .  
 Let  $Pi$  denote the IRD-partition in which  $Ei$  exists.

- (1) If *U* does not have the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES, then for each *Pi*, there shall be an IRD-view *Vi* such that the following are satisfied:
  - (a) *Vi* is related to *U*.
  - (b) *Vi* is associated with *Pi*.
  - (c) One of the following is satisfied:
    - (i) *Vi* has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
    - (ii) *Vi* has an IRD-PERMISSIONS attribute-group with the ENTITY-TYPE-NAME attribute equal to ALL or the assigned-access-name of *Ti* and the SECURE-ENTITY-PERMISSION attribute equal to YES.

If the above conditions are met, it is said that *U* is authorized SECURE-ENTITY-PERMISSION for entities of type *Ti* in *Pi*.

- (2) If *U* is authorized IRD-ADMINISTRATOR-PERMISSION, then *U* shall be authorized SECURE-ENTITY-PERMISSION to all entities except as excluded by General Rule (2) above.
- (3) Each *Ei* shall be visible to *U*. In particular:
  - (a) If *Ei* is not secured, no additional requirements need be satisfied.
  - (b) If *Ei* is assigned access-controllers *Ck*, then one of the following shall be satisfied:
    - (i) *U* or *Vi* has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
    - (ii) *Vi* has the read or write key for some access-controller *Ck*.
- (4) Each controller specified in related-controllers-clause shall be a member of *S*.

#### **Actions**

- (1) Syntax validation shall be performed. All appropriate error and warning messages shall be issued.
- (2) If any errors were encountered, the command shall be terminated.
- (3) Entities shall be selected. For each selected or specified entity, access-controllers shall be deassigned, according to General Rules (3) and (4). If an error is encountered in processing a given entity, all appropriate error and warning messages shall be issued for that entity. The command shall proceed to the next specified or selected entity.
- (4) End of command processing shall be confirmed.

#### **Error and Warning Conditions**

- (1) Error E01009: Entity does not exist or is not visible. See General Rule (1).
- (2) Error E03007: Cannot secure a security entity. See General Rule (2).

- (3) Warning W03005: Specified entity is still secured. It is assigned one or more access-controllers not in user's security maintenance set. See General Rule (3)(a).
- (4) Warning W03006: Specified entity is not secured. See General Rule (3)(b).
- (5) Warning W03007: Specified access-controller not assigned to entity. See General Rule (4).
- (6) Error E03008: Entity-level security cannot be assigned to specified entity. See General Rule (5).
- (7) Error E03011: User is not authorized to perform specified action(s) by effective IRD-view. See Security Rule (1) and Security Rule (3)(b)(ii).
- (8) Error E03004: User not authorized to perform specified action(s). (The user is not authorized SECURE-ENTITY-PERMISSION in any IRD partition.) See Security Rules (1) and (2).
- (9) Error E03025: User cannot assign/deassign specified access-controller. See Security Rule (4).

#### 6.4 Add Access-Key Command

**Function** To assign the read or write access-keys corresponding to one or more access-controllers to one or more IRD-views.

**Format**

```

add-access-key-command ::=
    add-word                M1.4.3
    read-or-write-option
    access-keys-word        5
    controller-selection-option
    to-word                  M1.4.3
    IRD-views-option
    ;

read-or-write-option ::=
    read-word                5
    | write-word             5

controller-selection-option ::=
    for-all-controllers-clause 7.7
    | controller-list-clause    7.5
    | for-controllers-in-clause 7.11

IRD-views-option ::=
    IRD-views-list-clause     7.6
    | IRD-views-in-clause    7.10
    
```

**Syntax Rules** None.

## General Rules

For the following General Rules:

Let  $V_i$ ,  $1 \leq i \leq n$ , denote an IRD-view either specified or selected via view-selection option.

Let  $C_j$ ,  $1 \leq j \leq m$ , denote an access-controller either specified or selected via controller-selection option.

- (1) If  $V_i$  has the access-key for  $C_j$  of the type identified by read-or-write-option, then no change of access-key shall take place and a warning shall be issued.
- (2) Each entity specified in or selected via controller-selection-option shall be an entity of type ACCESS-CONTROLLER.
- (3) Each entity specified in or selected via IRD-views-option shall be an entity of type IRD-VIEW.

## Security Rules

- (1) If one of the following is satisfied, the command shall be permitted to execute:
  - (a)  $U$  has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
  - (b)  $V$  has the attribute IRD-PARTITION-NAME equal to SECURITY and the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES.
  - (c)  $V$  has the attribute IRD-PARTITION-NAME equal to SECURITY, and  $V$  authorizes READ-PERMISSION to entities of type ACCESS-CONTROLLER and MODIFY-PERMISSION to entities of type IRD-VIEW.

In each of these cases, it is said that unrestricted ACCESS-KEYS-PERMISSION is authorized.

- (2) If  $V$  has the attribute IRD-PARTITION-NAME equal to SECURITY, and the attribute ACCESS-KEYS-PERMISSION equal to YES, then  $V$  is said to authorize restricted ACCESS-KEYS-PERMISSION. If  $V$  authorizes only restricted ACCESS-KEYS-PERMISSION, then each  $V_i$  as above shall be related to  $U$ .

## Actions

- (1) Syntax validation shall be performed. All appropriate error and warning messages shall be issued.
- (2) If any errors were encountered, the command shall be terminated.
- (3) Let  $k_j$  denote the access-key of the type specified by read-or-write-option for the corresponding  $C_j$ .

Each  $V_i$  shall be processed against each  $C_j$ . The order of processing shall not be significant. For example, it is possible to pass  $V_i$ 's against  $C_j$ 's or to pass  $C_j$ 's against  $V_i$ 's. An implementation may choose any such approach.

For each  $V_i, C_j$  pair,  $k_j$  shall be assigned to  $V_i$ . If an error is encountered in processing a given security entity, all appropriate error and warning messages shall be issued for that entity. The command shall proceed to the next specified or selected security entity, according to the processing strategy in place.

- (4) End of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E03020: Entity is not an access-controller. See General Rule (2).
- (2) Error E03021: Entity is not an IRD-view. See General Rule (3).
- (3) Error E03022: Command not authorized against specified or selected IRD-view. See Security Rule (2).
- (4) Error E03011: User not authorized to perform specified action(s) by the effective IRD-view. See Security Rules (1) and (2).

**6.5 Delete Access-Key Command**

**Function** To deassign all or selected access-keys for one or more IRD-views.

**Format**

```

delete-access-key-command ::=
    delete-word                M1.4.3
    access-keys-word           5
    controller-selection-option
    to-word                    M1.4.3
    IRD-views-option
    ;

controller-selection-option ::=
    for-all-controllers-clause 7.7
    | controller-list-clause    7.5
    | for-controllers-in-clause 7.11

IRD-views-option ::=
    IRD-views-list-clause      7.6
    | IRD-views-in-clause      7.10
    
```

**Syntax Rules** None.

**General Rules**

- (1) Each entity specified in or selected via IRD-views-option shall be an entity of type IRD-VIEW.
- (2) Each entity specified in or selected via controller-selection-option shall be an entity of type ACCESS-CONTROLLER.

- (3) If a selected or specified IRD-view does not have the corresponding access-key for a selected or specified access-controller, no action shall take place with respect to that IRD-view and access-controller.
- (4) If for-all-controllers-clause is specified, then for each IRD-view specified in or selected via view-selection-option, all access-keys assigned to that IRD-view shall be deleted.

### Security Rules

- (1) If one of the following is satisfied, the command shall be permitted to execute.
  - (a) *U* has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
  - (b) *V* has the attribute IRD-PARTITION-NAME equal to SECURITY and the IRD-ADMINISTRATOR-PERMISSION attribute equal to YES.
  - (c) *V* has the attribute IRD-PARTITION-NAME equal to SECURITY, and *V* authorizes READ-PERMISSION to entities of type ACCESS-CONTROLLER and MODIFY-PERMISSION to entities of type IRD-VIEW.

In each of these cases, it is said that unrestricted ACCESS-KEYS-PERMISSION is authorized.

- (2) If *V* has the attribute IRD-PARTITION-NAME equal to SECURITY, and the attribute ACCESS-KEYS-PERMISSION equal to YES, then *V* is said to authorize restricted ACCESS-KEYS-PERMISSION. If *V* authorizes only restricted ACCESS-KEYS-PERMISSION, then each *V<sub>i</sub>* as above shall be related to *U*.

### Actions

- (1) The command shall be validated. All appropriate error and warning messages shall be issued.
- (2) If any errors were encountered, the command shall be terminated.
- (3) Let *k<sub>j</sub>* denote the access-key of the type specified by read-or-write-option for the corresponding *C<sub>j</sub>*.

Each *V<sub>i</sub>* shall be processed against each *C<sub>j</sub>*. The order of processing shall not be significant. For example, it is possible to pass *V<sub>i</sub>*'s against *C<sub>j</sub>*'s or to pass *C<sub>j</sub>*'s against *V<sub>i</sub>*'s. An implementation may choose either approach.

For each *V<sub>i</sub>*, *C<sub>j</sub>* pair, *k<sub>j</sub>* shall be deassigned from *V<sub>i</sub>*. If an error is encountered in processing a given security entity, all appropriate error and warning messages shall be issued for that entity. The command shall proceed to the next specified or selected security entity, according to the processing strategy in place.

- (4) End of command processing shall be confirmed.

### Error and Warning Conditions

- (1) Error E03021: Entity is not an IRD-view. See General Rule (1).

- (2) Error E03020: Entity is not an access-controller. See General Rule (2).
- (3) Error E03022: Command not authorized against specified or selected IRD-view. See Security Rule (2).
- (4) Error E03011: User not authorized to perform specified action(s) by the effective IRD-view. See Security Rules (1) and (2).

## 7 IRDS Security Command-Clauses

The following command-clauses are defined for Module 3. Some command-clauses are unique to the IRDS Security Commands; others are required by modifications to Module 1.

### 7.1 Entities In Command-Clause

**Function** To identify a set of entities whose security is to be maintained.

**Format**

```

entities-in-clause ::=
    entities-word           M1.4.3
    in-word                 M1.4.3
    entity-selection-set-spec

entity-selection-set-spec ::=
    [ using-IRD-views-clause ]           M1.6.38
    entity-selection-criteria-clause     M1.6.33
    | entity-selection-criteria-clause   M1.6.33
    [ using-IRD-views-clause ]           M1.6.38
    
```

**Syntax Rules** None.

**General Rules**

- (1) Using-IRD-views-clause in this command shall be used to restrict the selection of entities. If no using-IRD-views-clause is specified, then "USING IRD-VIEWS = ALL" shall be used.
- (2) The only IRD-views which are meaningful in this command-clause are those which are not associated with the IRD-partition named SECURITY.
  - (a) If using-IRD-views-clause specifies ALL, then all IRD-views related to the effective IRDS-user which are not associated with the IRD-partition named SECURITY shall be used.
  - (b) If using-IRD-views-clause specifies an IRD-view which is associated with the IRD-partition named SECURITY, an error condition shall exist.

**Error and Warning Conditions**

- (1) Error E03026: Invalid IRD-view for command-clause. Specified IRD-view associated with SECURITY IRD-partition.

**7.2 From Controller Command-Clause**

**Function** To identify an existing access-controller for one or more entities.

**Format**

```

from-controller-clause ::=
    from-word                M1.4.3
    [ controller-word ]      5
    access-controller-name   5

```

**Syntax Rules** None.

**General Rules**

- (1) Access-controller-name shall be the assigned-access-name of an entity existing in the IRD.
- (2) Access-controller-name shall identify an entity of type ACCESS-CONTROLLER.

**Error and Warning Conditions**

- (1) Error E03023: Access-controller-name does not exist in the IRD. General Rule (1) has been violated.
- (2) Error E03020: Entity is not an access-controller. General Rule (2) has been violated. General Rule (1) has not been violated.

**7.3 To Controller Command-Clause**

**Function** To identify a replacement access-controller for one or more entities within a modify security command.

**Format**

```

to-controller-clause ::=
    to-word                M1.4.3
    [ controller-word ]    5
    access-controller-name  5

```

**Syntax Rules** None.

**General Rules**

- (1) Access-controller-name shall be the assigned-access-name of an entity existing in the IRD.



- (2) Access-controller-name shall identify an entity of type ACCESS-CONTROLLER.

**Error and Warning Conditions**

- (1) Error E03023: Access-controller-name does not exist in the IRD. General Rule (2) has been violated.
- (2) Error E03020: Entity is not an access-controller. General Rule (3) has been violated. General Rule (2) has not been violated.

**7.4 Related Controllers Command-Clause**

**Function** To identify those access-controllers which are to be disassociated from one or more entities via a delete security command.

**Format**

```

related-controllers-clause ::=
    controllers-word                    5
    =
    controller-identification

controller-identification ::=
    all-word                            M1.4.3
    | controller-name-list

controller-name-list ::=
    access-controller-name              5
    [ , controller-name-list ]
    
```

**Syntax Rules** None.

**General Rules**

- (1) Each access-controller-name shall be the assigned-access-name of an entity existing in the IRD.
- (2) Each access-controller-name shall identify an entity of type ACCESS-CONTROLLER.
- (3) All-word shall denote all access-controllers within the security maintenance set *S* for the effective IRDS-user. (See General Rule (2) of Section 6.) Thus if all-word is specified, all access-controllers within *S* shall be deassigned from the entity whose security is being maintained. Any access-controller assigned to the entity which is not a member of *S* shall not be deleted.

**Error and Warning Conditions**

- (1) Error E03023: Access-controller-name does not exist in the IRD. General Rule (1) has been violated.
- (2) Error E03020: Entity is not an access-controller. General Rule (2) has been violated. General Rule (1) has not been violated.

### 7.5 Controller-List Command-Clause

**Function** To identify corresponding access-controllers.

**Format**

```
controller-list-clause ::=
    controllers-word           5
    =
    controller-name-list
```

```
controller-name-list ::=
    access-controller-name    5
    [ , controller-name-list ]
```

**Syntax Rules** None.

**General Rules**

- (1) Each access-controller-name shall be the assigned-access-name of an entity existing in the IRD.
- (2) Each access-controller-name shall identify an entity of type ACCESS-CONTROLLER.

**Error and Warning Conditions**

- (1) Error E03023: Access-controller-name does not exist in the IRD. General Rule (1) has been violated.
- (2) Error E03020: Entity is not an access-controller. General Rule (2) has been violated. General Rule (1) has not been violated.

### 7.6 IRD-Views List Command-Clause

**Function** To identify those IRD-views for which access-keys are to be assigned or removed.

**Format**

```
IRD-views-list-clause ::=
    IRD-view-word             M1.4.3
    =
    IRD-view-name-list
```

```
IRD-view-name-list ::=
    IRD-view-name            5
    [ , IRD-view-name-list ]
```

**Syntax Rules** None.

**General Rules**

- (1) Each IRD-view-name shall be the assigned-access-name of an existing entity in the IRD.

- (2) Each IRD-view-name shall identify an entity of type IRD-VIEW.

**Error and Warning Conditions**

- (1) Error E03024: IRD-view-name does not exist in the IRD. General Rule (1) has been violated.
- (2) Error E03021: Entity is not an IRD-view. General Rule (2) has been violated. General Rule (1) has not been violated.

**7.7 For All Controllers Command-Clause**

**Function** Within add-security-command, to identify that access-keys for all controllers are to be assigned to the specified IRD-views. Within delete-security command, to identify that each access-key for every access-controller associated with a particular IRD-view shall be deleted.

**Format**

```

for-all-controllers-clause ::=
    for-word                M1.4.3
    all-word                M1.4.3
    controllers-word        5
    
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

**7.8 Assign Security Command-Clause**

**Function** To establish entity-level security for a newly created entity.

**Format**

```

assign-security-clause ::=
    assign-word              5
    security-word            5
    [ controller-word        5
      =
    ]
    access-controller-name   5
    
```

**Syntax Rules** None.

**General Rules**

- (1) Access-controller-name shall be the assigned-access-name of an entity existing in the IRD.
- (2) Access-controller-name shall identify an entity of type ACCESS-CONTROLLER.

### Error and Warning Conditions

- (1) Error E03023: Access-controller-name does not exist in the IRD. See General Rule (1).
- (2) Error E03020: Entity is not an access-controller. General Rule (2) has been violated. General Rule (1) has not been violated.

### 7.9 Exclude Security Command-Clause

**Function** When an entity has been created by either a modify-entity command or a copy-entity command, this command-clause identifies that the new entity shall not assume the security constraints of the original entity.

#### Format

```
exclude-security-clause ::=
    exclude-word           M1.4.3
    security-word         5
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

### 7.10 IRD-Views In Command-Clause

**Function** To identify a set of IRD-views to be used in an entity-level security maintenance command.

#### Format

```
IRD-views-in-clause ::=
    IRD-views-word       M1.4.3
    in-word              M1.4.3
    entity-selection-set-spec

entity-selection-set-spec ::=
    [ using-IRD-views-clause ] M1.6.38
    entity-selection-criteria-clause M1.6.33
    | entity-selection-criteria-clause M1.6.33
    [ using-IRD-views-clause ] M1.6.38
```

**Syntax Rules** None.

#### General Rules

- (1) The only IRD-views which shall be meaningful in this command-clause are those which are associated with the IRD-partition named SECURITY.
  - (a) If using-IRD-views-clause is not specified, the effective IRD-view shall be used.

- (b) If using-IRD-views-clause specifies ALL, then all IRD-views associated with the SECURITY IRD-partition shall be used.
- (c) If using-IRD-views-clause is specified, then only those IRD-views which are associated with the SECURITY IRD-partition shall be specified.

**Error and Warning Conditions**

- (1) Error E03027: Invalid IRD-view for command-clause. Specified IRD-view must be associated with SECURITY IRD-partition. See General Rule (1).
- (2) Error E01009: Entity does not exist or is not visible. See General Rule (2).

NOTE: The only entities which shall be meaningful to select by entity-election-criteria-clause are those of type IRDS-USER.

**7.11 For Controllers In Command-Clause**

**Function** To identify a set of access-controllers to be used in an entity-level security maintenance command.

**Format**

```

for-controllers-in-clause ::=
    for-word                M1.4.3
    controllers-word        5
    in-word                 M1.4.3
    entity-selection-set-spec

entity-selection-set-spec ::=
    [ using-IRD-views-clause ] M1.6.38
    entity-selection-criteria-clause M1.6.33
    | entity-selection-criteria-clause M1.6.33
    [ using-IRD-views-clause ] M1.6.38
    
```

**Syntax Rules** None.

**General Rules**

- (1) The only IRD-views which shall be meaningful in this command-clause are those which are associated with the IRD-partition named SECURITY.
  - (a) If using-IRD-views-clause is not specified, the effective IRD-view shall be used.
  - (b) If using-IRD-views-clause specifies ALL, then all IRD-views associated with the SECURITY IRD-partition shall be used.
  - (c) If using-IRD-views-clause is specified, then only those IRD-views which are associated with the SECURITY IRD-partition shall be specified.

**Error and Warning Conditions**

- (1) Error E01009: Entity does not exist or is not visible. See General Rule (2).

NOTE: The only entities which shall be meaningful to select by entity-selection-criteria-clause are those of type ACCESS-CONTROLLER.

## 8 Modifications to Module 1 Commands and Panel Trees

This section identifies those changes affected by this Module to the commands and panel trees defined in Module 1.

This section maintains a parallel organization with the command specifications of Module 1. If a change affects all the commands within a group (eg., "IRD Output Commands"), then the change shall be identified only once for the group. If a command is not explicitly identified, or it is not part of a group of commands for which changes have been identified, then there are no changes to that command.

For each affected command, only the modifications shall be identified.

- (1) Format
- (2) Syntax Rules
- (3) General Rules
- (4) Actions
- (5) Error and Warning Conditions

Unless otherwise stated, the Security Rules for each command or group of commands completely replace the corresponding Subsection in Module 1.

The only modification to the panel trees shall be the modifications to the data area which correspond to the modifications in the command language syntax. These are identified in detail at the end of Section 8.

### Notation for Identifying Modifications

Changes are identified as being in one of the following classes:

- (1) *Additions.* The additional wording or syntax does not require any changes to any other rules and/or actions. Additional syntax will be identified by a comment identifying the new syntax. Additional rules, actions, or error or warning conditions are identified by a "(+digit)" to indicate the relative position of the new text in the series.
- (2) *Modifications.* Syntax or a given rule and/or action has been changed. The overall sense of the rule or action, however, remains the same. For syntax, the change shall be minor.
- (3) *Replacements.* A given rule or the format has been substantially changed. The new rules are identified using the notation "(digit)" rather than the "(+digit)" notation for additions.
- (4) *Deletions.* A rule, action or syntax has been deleted. Any corresponding error/warning condition is also identified.

For error and warning conditions which correspond to additional rules the correspondence to the additional rule is identified using the relative number of the additional rule (i.e., "(+digit)" ).

## Security Rules Common to All Commands

The following Security Rules replace the Security Rules of Section 5 of Module 1.

- (1) The term "user" means the individual who requests the IRDS to perform one of the functions specified by the commands specified within this and subordinate subsections. Except as noted, it is assumed that the user has identified him/herself to the IRDS by the assigned-access-name of an IRDS-USER entity in the IRD. The mechanism for user identification shall be implementation-dependent. The specific IRDS-USER entity so identified is called the effective IRDS-user.
- (2) The following permissions shall be defined for each IRDS-USER entity:

```

COMMAND-LANGUAGE-PERMISSION
IRD-ADMINISTRATOR-PERMISSION
IRD-RENAME-PERMISSION
PANEL-PERMISSION
IRD-SCHEMA-ADMINISTRATOR-PERMISSION
IRD-SCHEMA-RENAME-PERMISSION

```

For each of these permissions, there is an attribute named as above. Each attribute has one of two possible values, YES or NO. There is a partial ordering of these permissions. In this partial ordering, the higher permission implies the lower permission. That is, given permissions  $P1$  and  $P2$  such that  $P1 > P2$ , if  $P1 = YES$ , then the IRDS also grants the permission defined by  $P2$ . If  $P1 = YES$  and  $P2 = NO$  or is undeclared, the higher permission dominates, i.e., explicit denial of a lower permission does not override the permissions granted at a higher level. Thus the phrase "is authorized xxx-PERMISSION" is taken to mean that either the corresponding attribute or a superior attribute equals YES.

The ordering among the above permissions is as follows:

```

IRD-ADMINISTRATOR-PERMISSION > IRD-RENAME-PERMISSION
IRD-ADMINISTRATOR-PERMISSION > COMMAND-LANGUAGE-PERMISSION
IRD-ADMINISTRATOR-PERMISSION > PANEL-PERMISSION
IRD-SCHEMA-ADMINISTRATOR-PERMISSION > IRD-SCHEMA-RENAME-PERMISSION
IRD-SCHEMA-ADMINISTRATOR-PERMISSION > COMMAND-LANGUAGE-PERMISSION
IRD-SCHEMA-ADMINISTRATOR-PERMISSION > PANEL-PERMISSION

```

- (3) In a system that has implementations of both the command language and the panel interface:
  - (a) In order to execute any command, the effective IRDS user shall be authorized COMMAND-LANGUAGE-PERMISSION and all other permissions required by the command.
  - (b) In order to perform the functionality of a command using the panel interface, the effective IRDS user shall be authorized PANEL-PERMISSION and all the other permissions required by the command.

**Additional Error and Warning Conditions**      None.

## 8.1 Modifications to IRD Schema Commands

All transactions which operate against the IRD Schema resulting from IRD Schema commands or their corresponding functions in the IRD Panel Interface shall be checked for conformance to security rules.

Permissions for any given IRDS-user to operate against the IRD Schema are defined at two levels. An entity of type IRDS-USER defines those permissions which apply globally for an IRDS-user. Entities of type IRD-SCHEMA-VIEW which are related to a given IRDS-user define permissions available within a given IRD Schema life-cycle-phase.

Each IRD-schema-view defines permissions to read, add, modify, and delete meta-entities and meta-relationships by their respective types. The permissions for meta-entities can be defined for each type or for all types. For meta-relationships, permissions are assumed based on meta-entity-type permissions unless the meta-relationship-type is specifically excluded.

The effective IRDS-user can have zero, one, or many IRD-schema-views. At any one time, one such IRD-schema-view shall be used to determine permissions to perform IRD Schema maintenance. This IRD-SCHEMA-VIEW entity is called the effective IRD-schema-view or the IRD-schema-view in effect. After completion of sign-on, the default IRD-schema-view for the effective IRDS-user shall become the effective IRD-schema-view.

The following notation is used in defining new rules for all IRD Schema commands in this Module:

- Let  $U$  denote the effective IRDS-user.
- Let  $V$  denote the effective IRD-schema-view.
- Let  $P$  denote the IRD Schema life-cycle-phase associated with  $V$ .

### Security Rules

- (+1) In order to execute any IRD Schema command,  $U$  shall have at least one associated IRD-SCHEMA-VIEW entity. This is defined in the IRD by an entity of type IRD-SCHEMA-VIEW which is related to the effective IRDS-user by a relationship of type IRDS-USER-HAS-IRD-SCHEMA-VIEW.
- (+2) IRD-SCHEMA-VIEW entities shall define the permissions authorized to a user within a given IRD Schema life-cycle-phase. This life-cycle-phase is identified by the attribute IRD-SCHEMA-PHASE-NAME, which shall exist and be equal to one of the following values: UNCONTROLLED, CONTROLLED, ARCHIVED.
- (+3) The effective IRDS-user can have multiple IRD-SCHEMA-VIEWS. For updating the IRD Schema, only one of the IRD-SCHEMA-VIEW entities shall be in effect at any one time. This is called the effective IRD-schema-view,  $V$ .
- (+4) The user can change the effective IRD-schema-view by using the set-session-defaults command (See Module 1, Subsection 5.3.2.1).
- (+5) The IRD-SCHEMA-VIEW contains the attribute-type IRD-SCHEMA-ADMINISTRATOR-PERMISSION and two repeating attribute-group-types, EXCLUDE-META-RELATIONSHIPS and IRD-SCHEMA-PERMISSIONS.



The attribute-group-type IRD-SCHEMA-PERMISSIONS has five component attribute-types:

META-ENTITY-TYPE-NAME  
 READ-PERMISSION  
 ADD-PERMISSION  
 MODIFY-PERMISSION  
 DELETE-PERMISSION

Within groups of the type IRD-SCHEMA-PERMISSIONS each "-PERMISSION" attribute identifies whether or not the identified action shall be authorized for the meta-entity-type identified in the corresponding META-ENTITY-TYPE-NAME attribute. Within any IRD-schema-view, there can exist only one IRD-SCHEMA-PERMISSIONS attribute-group for each unique META-ENTITY-TYPE-NAME attribute.

Within each group, permissions are ordered as follows:

DELETE-PERMISSION > MODIFY-PERMISSION > ADD-PERMISSION >  
 READ-PERMISSION

In any given IRD-schema-view, there can exist one IRD-SCHEMA-PERMISSIONS attribute-group with META-ENTITY-TYPE-NAME = ALL. This defines the minimum set of permissions which apply to all meta-entities of all types within the associated IRD Schema life-cycle-phase. A permission authorized for a specific meta-entity-type shall override a denial of permission specified in the IRD-SCHEMA-PERMISSIONS defined for all meta-entity-types. Conversely, a permission authorized for all entity-types shall not be denied by specifying NO in the corresponding permission for a specific type.

Within a given IRD-schema-view, IRD-SCHEMA-ADMINISTRATOR-PERMISSION shall be superior to all IRD-SCHEMA-PERMISSIONS.

- (+6) Permissions defined for a given IRDS-user entity and permissions defined within related IRD-schema-views are ordered as follows:

IRD-SCHEMA-ADMINISTRATOR-PERMISSION for an IRDS-user > IRD-SCHEMA-ADMINISTRATOR-PERMISSION for each related IRD-schema-view.

IRD-SCHEMA-RENAME-PERMISSION for an IRDS-user >  
 IRD-SCHEMA-PERMISSIONS = ( META-ENTITY-TYPE-NAME = ALL,  
 READ-PERMISSION = YES, ADD-PERMISSION = NO, MODIFY-  
 PERMISSION = NO, DELETE-PERMISSION = NO )

for each related IRD-schema-view

- (+7) Let "xxx" denote one of the following: READ, ADD, MODIFY, or DELETE. An IRD-schema-view  $V_1$  with an IRD-SCHEMA-PHASE-NAME attribute equal to  $P'$  is said to authorize xxx-PERMISSION for meta-entities of type  $T$  in  $P'$  if one of the following is satisfied:

- (a)  $V_1$  is related to  $U$  and  $U$  authorizes IRD-SCHEMA-ADMINISTRATOR-PERMISSION.  
 (b)  $V_1$  has the attribute IRD-SCHEMA-ADMINISTRATOR-PERMISSION equal to YES.

- (c) *V1* has an IRD-SCHEMA-PERMISSIONS attribute-group with component attributes META-ENTITY-TYPE-NAME equal to ALL or *T* and xxx-PERMISSION or a superior permission attribute equal to YES.

If one of the above conditions is satisfied, a meta-entity of type *T* is said to be visible in *V1*.

For Security Rules (+8) through (+11) inclusive:

Let *E1* and *E2* denote meta-entities.

Let *T1* and *T2* denote the type of *E1* and *E2*, respectively.

Let *P1* and *P2* denote the IRD Schema life-cycle-phases which contain *E1* and *E2*, respectively.

Let *Tr* be a meta-relationship-type.

Let *E1 Tr E2* be a meta-relationship of type *Tr*.

- (+8) An IRD-schema-view *V1* with an IRD-SCHEMA-PHASE-NAME attribute equal to *P1* is said to authorize READ-PERMISSION for meta-relationships of type *Tr* if one of the following is satisfied:

- (a) *V1* is related to *U* and *U* authorizes IRD-SCHEMA-ADMINISTRATOR-PERMISSION.

- (b) *V1* has the attribute IRD-SCHEMA-ADMINISTRATOR-PERMISSION equal to YES.

- (c) *V1* satisfies each of the following:

- (i) *V1* authorizes READ-PERMISSION to at least one meta-entity-type which participates in *Tr*.

- (ii) *V1* has no EXCLUDE-META-RELATIONSHIPS attribute equal to *Tr*.

A given meta-relationship *E1 Tr E2* of type *Tr* is said to be visible in a view *V1* if *V1* authorizes READ-PERMISSION for meta-relationships of type *Tr*. If both *E1* and *E2* are visible in *V1* then *E1 Tr E2* is said to be completely visible in *V1*; otherwise the meta-relationship is said to be partially visible in *V1*.

Note that even though a meta-relationship is only partially visible by any given IRD-schema-view, it can be completely visible to an IRDS-user. This can be achieved through two IRD-schema-views *V1* and *V2*. The following Security Rule covers this case.

- (+9) A meta-relationship shall be completely visible to *U* by two IRD-schema-views *V1* and *V2* if each of the following is satisfied:

- (a) Both *V1* and *V2* are related to *U*.

- (b) *E1* is visible in *V1*.

- (c) *E2* is visible in *V2*.

- (d) *E1 Tr E2* is partially visible in either *V1* or *V2* or both.

For Security Rules (+10) and (+11):

Let  $P1$  be the IRD Schema life-cycle-phase associated with  $V1$ .  
 Let  $P2$  be the IRD Schema life-cycle-phase associated with  $V2$ .

Rather than defining separate ADD, MODIFY, and DELETE permissions for relationships, only UPDATE-PERMISSION is defined. (This is because any change in a meta-relationship can have an impact on the first participating meta-entity.)

Authorization of UPDATE-PERMISSION for meta-relationships is defined for two distinct cases:

- (a)  $P1 = P2$ . This case is covered by Security Rule (+10).
  - (b)  $P1$  and  $P2$  are different. This case is covered by Security Rule (+11).
- (+10) If  $P1$  equals  $P2$ ,  $V1$  is said to authorize UPDATE-PERMISSION for the meta-relationship  $E1$  Tr  $E2$  if each of the following is satisfied:
- (a)  $V1$  authorizes MODIFY-PERMISSION for  $E1$ .
  - (b)  $V1$  authorizes READ-PERMISSION to  $E2$ .
  - (c) At least one of the following is satisfied:
    - (i)  $V1$  does not contain an attribute of type EXCLUDE-META-RELATIONSHIPS equal to  $Tr$ .
    - (ii)  $V1$  authorizes IRD-SCHEMA-ADMINISTRATOR-PERMISSION.
- (+11) Assume that  $P1$  and  $P2$  are different. Then  $V1$  and  $V2$  authorize UPDATE-PERMISSION for  $E1$  Tr  $E2$  if each of the following is satisfied:
- (a) Both  $V1$  and  $V2$  are related to  $U$ .
  - (b)  $V1$  authorizes MODIFY-PERMISSION for  $E1$ .
  - (c)  $V2$  authorizes READ-PERMISSION to  $E2$ .
  - (d) At least one of the following is satisfied:
    - (i)  $U$  authorizes IRD-SCHEMA-ADMINISTRATOR-PERMISSION.
    - (ii)  $V1$  authorizes IRD-SCHEMA-ADMINISTRATOR-PERMISSION.
    - (iii)  $V1$  does not contain an attribute of type EXCLUDE-META-RELATIONSHIPS equal to  $Tr$ .

#### Additional Error Messages

- (+1) Error E03001: IRDS-user has no corresponding IRD-schema-view. See Security Rule (+1).

### 8.1.1 Modifications to IRD Schema Maintenance Commands

The following IRD Schema Maintenance commands are affected when the facilities of this Module are in effect:

```
add-meta-entity-command
modify-meta-entity-command
delete-meta-entity-command
add-meta-relationship-command
modify-meta-relationship-command
delete-meta-relationship-command
modify-meta-entity-access-name-command
modify-meta-entity-descriptive-name-command
modify-meta-entity-life-cycle-phase-command
copy-meta-entity-command
deactivate-IRD-command
restore-IRD-schema-command
activate-IRD-command
```

The rules which define permissions for the different kinds of updates to the IRD Schema are defined within the Security Rules for these commands.

#### 8.1.1.1 Modifications to Add Meta-Entity Command

##### Security Rules

- (+1) The command shall be permitted to execute if *V* authorizes ADD-PERMISSION or if *U* is authorized IRD-SCHEMA-ADMINISTRATOR-PERMISSION.

##### Additional Error and Warning Conditions

- (+1) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (+1).

#### 8.1.1.2 Modifications to Modify Meta-Entity Command

##### Security Rules

For the following Security Rules:

Let *E* denote the specified meta-entity.  
Let *T* denote the type of the meta-entity *E*.  
Let *P1* denote the IRD Schema life-cycle-phase in which *E* exists.

- (+1) If new-meta-entity-version-clause is not specified, and *V* has the attribute IRD-SCHEMA-PHASE-NAME equal to *P*, then each of the following shall be satisfied:
- (a) If *T* is a type for which the modify meta-entity life-cycle-phase applies, *P1* and *P* shall be equal to UNCONTROLLED.
  - (b) At least one of the following shall be satisfied:
    - (i) *V* authorizes MODIFY-PERMISSION for meta-entities of type *T* in *P*.

- (ii)  $U$  authorizes IRD-SCHEMA-ADMINISTRATOR-PERMISSION.

This Rule is subject to the constraints imposed by General Rules (3) through (5) of Module 1, Subsection 5.1.1.2.

- (+2) If new-meta-entity-version-clause is specified, each of the following shall be satisfied:
- (a)  $V$  has the IRD-SCHEMA-PHASE-NAME attribute equal to UNCONTROLLED, and  $V$  authorizes ADD-PERMISSION for all meta-entities of the type  $T$ .
  - (b) If  $P1$  is the CONTROLLED or ARCHIVED IRD Schema life-cycle-phase, then  $U$  is related to at least one meta-entity of type IRD-SCHEMA-VIEW, which authorizes READ-PERMISSION to meta-entities of the type  $T$  in the IRD Schema life-cycle-phase  $P$ .
  - (c) Let  $E Ri Ej$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ , denote all meta-relationships in which  $E$  is the first meta-entity. Let  $E'$  denote the meta-entity to be created by this command. For each such  $E' Ri Ej$ , each of the following is satisfied:
    - (i) Either  $V$  authorizes UPDATE-PERMISSION for  $E' Ri Ej$ , or there exists another IRD-schema-view  $V^i$  so that  $V$  and  $V^i$  authorize UPDATE-PERMISSION.
    - (ii)  $E' Ri Ej$  does not violate IRD-schema-life-cycle-phase integrity.

#### Additional Error and Warning Conditions

- (+1) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rules (+1) and (+2)(a).
- (+2) Error E01067: Meta-entity does not exist or is not visible. See Security Rule (+2)(b).
- (+3) Warning W03001: User not authorized to create corresponding meta-relationship. See Security Rule (+2)(c)(i).
- (+4) Warning W01037: Creation of corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity. See Security Rule (+2)(c)(ii).

#### 8.1.1.3 Modifications to Delete Meta-Entity Command

##### Security Rules

For the following rules:

Let  $E$  denote the meta-entity to be deleted.

Let  $T$  denote the type of  $E1$ .

Let  $E Ri Ej$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ , denote all the meta-relationships in which  $E$  is the first meta-entity.

- (+1)  $V$  shall authorize DELETE-PERMISSION for meta-entities of type  $T$ .
- (+2) If with-meta-relationships-clause is specified, then for each  $E Ri Ej$ , one of the following shall be satisfied:

- (a)  $V$  authorizes UPDATE-PERMISSION for  $E$   $R_i$   $E_j$ .
- (b) There exists an IRD-schema-view,  $V_i$ , such that  $V$  and  $V_i$  authorize UPDATE-PERMISSION for  $E$   $R_i$   $E_j$ .

#### Additional Error and Warning Conditions

- (+1) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (+1).
- (+2) Error E03003: User not authorized to delete a meta-relationship in which specified meta-entity participates. See Security Rule (+2).

#### 8.1.1.4 Modifications to Add Meta-Relationship Command

##### Security Rules

For the following rules:

Let  $Tr$  denote the type of the specified meta-relationship.  
Let  $E1$  denote the first meta-entity in the meta-relationship.  
Let  $E2$  denote the second meta-entity in the meta-relationship.  
Let  $T1$  and  $T2$  denote the types of  $E1$  and  $E2$  respectively.  
Let  $P1$  denote the IRD Schema life-cycle-phase in which  $E1$  exists.  
Let  $P2$  denote the IRD Schema life-cycle-phase in which  $E2$  exists.  
Let  $E1$   $Tr$   $E2$  denote the meta-relationship to be added.

- (+1) If  $P1 = P2$ ,  $V$  shall authorize UPDATE-PERMISSION for the meta-relationship  $E1$   $Tr$   $E2$  as specified in Security Rule (+10) at the beginning of Section 8.
- (+2) If  $P1$  is not equal to  $P2$ , there shall be an IRD-schema-view  $V'$  related to  $U$  such that  $V$  and  $V'$  authorize ADD-PERMISSION for  $E1$   $Tr$   $E2$  as specified in Security Rule (+11) at the beginning of Section 8.

#### Additional Error and Warning Conditions

- (+1) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (+1).
- (+2) Error E03004: User not authorized to perform specified action. See Security Rule (+2).

#### 8.1.1.5 Modifications to Modify Meta-Relationship Command

##### Security Rules

For the following rules:

Let  $Tr$  denote the type of the specified meta-relationship.  
Let  $E1$  denote the first meta-entity in the meta-relationship.  
Let  $E2$  denote the second meta-entity in the meta-relationship.  
Let  $T1$  and  $T2$  denote the types of  $E1$  and  $E2$ , respectively.  
Let  $P1$  denote the IRD Schema life-cycle-phase in which  $E1$  exists.  
Let  $P2$  denote the IRD Schema life-cycle-phase in which  $E2$  exists.  
Let  $E1$   $Tr$   $E2$  denote the meta-relationship to be added.

- (+1) If  $P1 = P2$ ,  $V$  shall authorize MODIFY-PERMISSION for the meta-relationship  $E1 Tr E2$  as specified in Security Rule (+10) at the beginning of Section 8.
- (+2) If  $P1$  is not equal to  $P2$ , there shall be an IRD-schema-view  $V'$  related to  $U$  such that  $V$  and  $V'$  authorize MODIFY-PERMISSION for  $E1 Tr E2$  as specified in Security Rule (+11) at the beginning of Section 8.

#### **Additional Error and Warning Conditions**

- (+1) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (+1).
- (+2) Error E03004: User not authorized to perform specified action. See Security Rule (+2).

#### **8.1.1.6 Modifications to Delete Meta-relationship Command**

##### **Security Rules**

Let  $Tr$  denote the type of the specified meta-relationship.  
 Let  $E1$  denote the first meta-entity in the meta-relationship.  
 Let  $E2$  denote the second meta-entity in the meta-relationship.  
 Let  $T1$  and  $T2$  denote the types of  $E1$  and  $E2$ , respectively.  
 Let  $P1$  denote the IRD Schema life-cycle-phase in which  $E1$  exists.  
 Let  $P2$  denote the IRD Schema life-cycle-phase in which  $E2$  exists.  
 Let  $E1 Tr E2$  denote the meta-relationship to be added.

- (+1) If  $P1 = P2$ ,  $V$  shall authorize MODIFY-PERMISSION for the meta-relationship  $E1 Tr E2$  as specified in Security Rule (+10) at the beginning of Section 8.
- (+2) If  $P1$  is not equal to  $P2$ , there shall be an IRD-schema-view  $V'$  related to  $U$  such that  $V$  and  $V'$  authorize MODIFY-PERMISSION for  $E1 Tr E2$  as specified in Security Rule (+11) at the beginning of Section 8.
- (+3) If the IRD-SCHEMA-PHASE-NAME attribute in  $V$  equals ARCHIVED,  $V$  shall authorize DELETE-PERMISSION for meta-entities of type  $T1$ .

#### **Additional Error Conditions**

- (+1) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (+1).
- (+2) Error E03004: User not authorized to perform specified action. See Security Rule (+2).

#### **8.1.1.7 Modifications to Modify Meta-Entity Access-Name Command**

##### **Security Rules**

- (+1)  $U$  shall be authorized IRD-SCHEMA-RENAME-PERMISSION.

#### **Additional Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action. See Security Rule (+1).

### 8.1.1.8 Modifications to Modify Meta-Entity Descriptive-Name Command

#### Security Rules

(+1) *U* shall be authorized IRD-SCHEMA-RENAME-PERMISSION.

#### Additional Error and Warning Conditions

(+1) Error E03004: User not authorized to perform specified action. See Security Rule (+1).

### 8.1.1.9 Modifications to Modify Meta-Entity Life-Cycle-Phase Command

#### Security Rules

(+1) *U* shall be authorized IRD-SCHEMA-ADMINISTRATOR-PERMISSION.

#### Additional Error and Warning Conditions

(+1) Error E03004: User not authorized to perform specified action. See Security Rule (+1).

### 8.1.1.10 Modifications to Copy Meta-Entity Command

#### Security Rules

For the following Security Rules:

Let *E* denote the meta-entity to be copied.  
Let *E'* denote the meta-entity to be created via the copy command.  
Let *T* denote the type of *E*, and *E'*.  
Let *P1* denote the IRD Schema life-cycle-phase in which *E* exists.  
Let *R<sub>i</sub>*,  $1 \leq i \leq n$ , denote meta-relationship-types.  
Let *E R<sub>i</sub> E<sub>j</sub>*,  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ , denote all meta-relationships in which *E* is the first meta-entity.

- (+1) *V* shall authorize ADD-PERMISSION for entities of type *T* in *P*.
- (+2) If  $P \neq P1$ , then there shall exist another IRD-schema-view *V1* which is related to *U* and authorizes READ PERMISSION for meta-entities of type *T* in *P1*.
- (+3) If with-meta-relationships-clause is specified, then for each existing meta-relationship of the form *E R<sub>i</sub> E<sub>j</sub>* a new meta-relationship *E' R<sub>i</sub> E<sub>j</sub>* shall be created. For each such *E' R<sub>i</sub> E<sub>j</sub>*, each of the following shall be satisfied:
  - (a) One of the following is satisfied:
    - (i) *V* authorizes UPDATE-PERMISSION
    - (ii) There exists another IRD-schema-view *V<sub>i</sub>* so that *V* and *V<sub>i</sub>* authorize UPDATE-PERMISSION for *E' R<sub>i</sub> E<sub>j</sub>*.
  - (b) *E' R<sub>i</sub> E<sub>j</sub>* does not violate IRD-schema-life-cycle-phase integrity.



**Additional Error and Warning Conditions**

- (+1) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (+1).
- (+2) Error E01067: Meta-entity does not exist or is not visible. See Security Rule (+2).
- (+3) Warning W03001: User not authorized to create corresponding meta-relationship. See Security Rule (+3)(a).
- (+4) Warning W01037: Creation of corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity. See Security Rule (+3)(b).

**8.1.1.11 Modifications to Deactivate IRD Command****Security Rules**

- (+1) *U* shall be authorized IRD-SCHEMA-ADMINISTRATOR-PERMISSION.

**Additional Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action. See Security Rule (+1).

**8.1.1.12 Modifications to Restore IRD Schema Command****Security Rules**

- (+1) *U* shall be authorized IRD-SCHEMA-ADMINISTRATOR-PERMISSION.

**Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action. See Security Rule (+1).

**8.1.1.13 Modifications to Activate IRD Command****Security Rules**

- (+1) *U* shall be authorized IRD-SCHEMA-ADMINISTRATOR-PERMISSION.

**Additional Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action. See Security Rule (+1).

**8.1.2 Modifications to Output IRD Schema Command****Security Rules**

- (+1) If no using-IRD-schema-views-clause is specified, the only meta-entities which are visible shall be those which are visible in *V*.
- (+2) If using-IRD-schema-views-clause is specified, a meta-entity shall be visible if and only if it is visible in at least one of the IRD-schema-views specified in using-IRD-schema-views-clause.

- (+3) The IRD-schema-output command shall display only those meta-entities and meta-relationships which are visible according to the following rules:
- (a) Any meta-entity which fulfills the selection criteria specified command, but is not visible according to the above rules shall be ignored.
  - (b) Any meta-relationship in which a selected meta-entity participates which is not visible according to the above rules shall be ignored on the display.
  - (c) If a meta-relationship is partially visible, then the meta-relationship shall be displayed with a fill-character (X) in place of the invisible meta-entity name. Both the access-name and descriptive-name shall be replaced by fill-characters. Each component of these names shall be padded with fill-characters to the maximum length for the component.

**Additional Error and Warning Conditions**            None.

## 8.2 Modifications to IRD Commands and Panel Trees

All transactions which operate against the IRD resulting from IRD commands or their corresponding functions in the IRD Panel Interface shall be checked for conformance to security rules.

Permissions for any given IRDS-user are defined at two different levels. An Entity of type IRDS-USER defines permissions for a user which apply globally. Entities of type IRD-VIEW which are related to a given IRDS-user define permissions available within a given IRD-partition.

Each IRD-view defines permissions to read, add, modify, and delete entities and relationships by their respective types. The permissions for entities are defined by each type or for all types. Additional read and write restrictions can exist for individual entities. The facility which controls this is called entity-level security, and entities with such read or write restrictions are called secured entities. A given IRD-view can allow read or write access to individual secured entities.

For a relationship, permissions shall be assumed based on entity-type and participating entity permissions unless the corresponding relationship-type is specifically excluded.

The effective IRDS-user can have zero, one or many IRD-views. At any one time, one IRD-view shall be used to determine permissions to perform IRD maintenance. This IRD-VIEW entity is called the effective IRD-view or the IRD-view in effect. After completion of sign-on, the default IRD-view for the effective IRDS-user shall become the effective IRD-view.

### Security Rules

For all IRD commands and panel trees:

Let  $U$  denote the effective IRDS-user.  
Let  $V$  denote the effective IRD-view.  
Let  $P$  denote the IRD partition associated with  $V$ .

- (+1) In order to execute any IRD command,  $U$  shall have at least one associated IRD-VIEW entity. This is defined in the IRD by an entity of type IRD-VIEW which is related to  $U$  by a relationship of type IRDS-USER-HAS-IRD-VIEW.

- (+2) IRD-VIEW entities define the permissions authorized to a user within a given IRD partition by means of the attribute IRD-PARTITION-NAME. The value of the IRD-PARTITION-NAME attribute shall be equal to the assigned-access-name of a meta-entity of type IRD-PARTITION in the IRD Schema.
- (+3) The effective IRDS-user can have multiple IRD-VIEWS. For updating the IRD, only one of the IRD-VIEW entities shall be in effect at any one time. This is called the effective IRD-view, V. A default effective IRD-view shall be determined by a relationship of type IRDS-USER-HAS-IRD-VIEW.
- (+4) The user can change the effective IRD-view by using the set-session-defaults command (see Module 1, Subsection 5.3.2.1).
- (+5) The entity-type IRD-VIEW shall be associated with the following new attribute-types: IRD-ADMINISTRATOR-PERMISSION, ACCESS-KEYS-PERMISSION, READ-KEY, WRITE-KEY, and EXCLUDE-RELATIONSHIPS. The attribute-types ACCESS-KEYS-PERMISSION, READ-KEY and WRITE-KEY support IRDS entity-level security. They will be discussed in detail in subsequent rules governing entity-level security. The attribute-type EXCLUDE-RELATIONSHIPS can repeat. It identifies those relationship-types for which no permissions shall be granted.

The IRD-VIEW entity-type also contains an additional attribute-group-type, IRD-PERMISSIONS. This attribute-group-type shall be associated with the following component attribute-types:

```

ENTITY-TYPE-NAME
READ-PERMISSION
ADD-PERMISSION
MODIFY-PERMISSION
DELETE-PERMISSION
MODIFY-PHASE-PERMISSION
ADD-AND-SECURE-PERMISSION
SECURE-ENTITY-PERMISSION
    
```

Within groups of type IRD-PERMISSIONS, each "-PERMISSION" attribute identifies whether or not the identified action shall be authorized for the entity-type identified by the corresponding ENTITY-TYPE-NAME attribute.

Within each group, permissions are ordered as follows:

```

DELETE-PERMISSION > MODIFY-PERMISSION > ADD-PERMISSION >
                                                                READ-PERMISSION
ADD-AND-SECURE-PERMISSION > ADD-PERMISSION
SECURE-ENTITY-PERMISSION > READ-PERMISSION
MODIFY-PHASE-PERMISSION > READ-PERMISSION
    
```

Within any IRD-view, there shall exist only one IRD-PERMISSIONS attribute-group for each ENTITY-TYPE-NAME attribute. There can exist one IRD-PERMISSIONS attribute-group with ENTITY-TYPE-NAME = ALL. This defines the minimum set of permissions which apply to all entities of all types within the associated IRD-partition. A permission authorized for a specific entity-type shall override denial of permission specified in the IRD-PERMISSIONS defined for all entity-types. Conversely, specifying NO in a permission for a specific type shall not deny the corresponding permission authorized for all entity-types.

Within a given IRD-view, IRD-ADMINISTRATOR-PERMISSION shall be superior to all IRD-PERMISSIONS.

ACCESS-KEYS-PERMISSION is related to two IRD-PERMISSIONS attribute-group-types as follows:

ACCESS-KEYS-PERMISSION > IRD-PERMISSIONS =  
(ENTITY-TYPE-NAME = IRD-VIEW, READ = YES),  
(ENTITY-TYPE-NAME = ACCESS-CONTROLLER, READ = YES)

ACCESS-KEYS-PERMISSION and attribute-groups in which ENTITY-TYPE-NAME identifies a security entity-type are meaningful only if IRD-PARTITION-NAME = SECURITY.

- (+6) Permissions for a given IRDS-user and permissions defined within related IRD-views shall be ordered as follows:

IRD-ADMINISTRATOR-PERMISSION for an IRDS-user >  
IRD-ADMINISTRATOR-PERMISSION for each related IRD-view

IRD-RENAME-PERMISSION for an IRDS-user >  
IRD-PERMISSIONS = (ENTITY-TYPE-NAME = ALL, READ-PERMISSION = YES)

for each related IRD-view.

- (+7) An entity *E* of type *T* is said to be visible in an IRD-view *V1* if the following are satisfied:

- (a) The entity shall be in the IRD-partition identified by the IRD-PARTITION-NAME attribute of *V1*.
- (b) One of the following shall be satisfied:
  - (i) *V1* is related to *U* and *U* authorizes either IRD-ADMINISTRATOR-PERMISSION or IRD-RENAME-PERMISSION.
  - (ii) *V1* is authorized IRD-ADMINISTRATOR-PERMISSION.
  - (iii) *V1* has an IRD-PERMISSIONS attribute-group in which ENTITY-TYPE-NAME = ALL or the assigned-access-name of *T*, and either READ-PERMISSION or a superior permission attribute is equal to YES.
- (c) If (b)(iii) above is true and (b)(i) and (b)(ii) are not true, one of the following shall be satisfied:
  - (i) *E* is not secured, i.e., *E* has not been assigned any access-controllers.
  - (ii) If *E* is secured, then *V1* has a read-key or a write-key corresponding to at least one access-controller assigned to *E*.

The phrase "*V1* authorizes READ-PERMISSION for *E*" is equivalent to the phrase "*E* is visible in *V1*". The phrase "*V1* authorizes READ-PERMISSION for entities of type *T*" is defined by (a) and (b) above. Note that entity-level security allows for the possibility

that an entity  $E$  of type  $T$  is not visible in  $V1$  even though  $V1$  authorizes READ-PERMISSION for entities of type  $T$ .

- (+8) Let  $xxx$  denote one of the following: ADD, MODIFY, DELETE, MODIFY-LIFE-CYCLE-PHASE, ADD-AND-SECURE, SECURE-ENTITY. Let  $T$  denote an entity-type.  
Let  $V1$  be an IRD-view which is associated with IRD-partition  $P'$ .

$V1$  shall authorize  $xxx$ -PERMISSION to entities of type  $T$  in  $P'$  if one of the following is satisfied:

- (a)  $V1$  is related to  $U$  and  $U$  authorizes a permission which is superior to  $xxx$ -PERMISSION, and applies to entities of type  $T$ .
- (b)  $V1$  authorizes a permission which is superior to  $xxx$ -PERMISSION and applies to entities of type  $T$ .
- (c)  $V1$  has an IRD-PERMISSIONS attribute-group with component attributes ENTITY-TYPE-NAME equal to ALL or the assigned-access-name of  $T$  and  $xxx$ -PERMISSION or a superior permission attribute equals YES.

- (+9) Let  $T$ ,  $P'$  and  $V1$  be as in Security Rule (+8) above.

Let  $E$  be an entity of type  $T$ .  
Let  $yyy$  denote either MODIFY or DELETE.

$V1$  shall authorize  $yyy$ -PERMISSION for  $E$  if  $V1$  authorizes  $yyy$ -PERMISSION for entities of type  $T$ . If  $E$  is secured, then  $V1$  shall have a write-key corresponding to one of the access-controllers assigned to  $E$ . (Note that visibility of  $E$  is implied for both secured and unsecured entities.)

For any other permission identified in Security Rule (+8), the authorization of a permission for a specified entity is defined within the Security Rules for the relevant IRD commands.

For Security Rules (+10) through (+14) inclusive:

Let  $E1$  and  $E2$  denote entities.  
Let  $T1$  and  $T2$  denote the type of  $E1$  and  $E2$ , respectively.  
Let  $P1$  and  $P2$  denote the IRD partition which contain  $E1$  and  $E2$ , respectively.  
Let  $Tr$  be a relationship-type.  
Let  $E1 Tr E2$  be a relationship of type  $Tr$ .

- (+10) An IRD-view  $V1$  with an IRD-PARTITION-NAME attribute equal to  $P1$  shall authorize READ-PERMISSION for relationships of type  $Tr$  if one of the following is satisfied:
- (a)  $V1$  is related to  $U$  and  $U$  authorizes either IRD-ADMINISTRATOR-PERMISSION or IRD-RENAME-PERMISSION.
  - (b)  $V1$  has the attribute IRD-ADMINISTRATOR-PERMISSION equal to YES.
  - (c)  $V1$  satisfies each of the following:
    - (i)  $V1$  authorizes READ-PERMISSION for entities of type  $T1$  or  $T2$ .

- (ii) *V1* has no EXCLUDE-RELATIONSHIPS attribute equal to the assigned-access-name of *Tr*.

(+11) A given relationship *E1 Tr E2* of type *Tr* is said to be visible in a view *V1* if both of the following are satisfied:

- (a) Either *E1* or *E2* are visible in *V1*.
- (b) *V1* authorizes READ-PERMISSION for relationships of type *Tr*.

If both *E1* and *E2* are visible in *V1*, then *E1 Tr E2* is said to be completely visible in *V1*. If only one of these entities are visible in *V1*, then the relationship is said to be partially visible in *V1*.

Note that even though a relationship *E1 Tr E2* of type *Tr* is only partially visible by any given IRD-view, it can be completely visible to the IRDS-user. This can be achieved through two IRD-views *V1* and *V2*. The following Security Rule covers this case.

(+12) A relationship shall be completely visible to *U* by two IRD-views *V1* and *V2* if each of the following is satisfied:

- (a) Both *V1* and *V2* are related to *U*.
- (b) *E1* is visible in *V1*.
- (c) *E2* is visible in *V2*.
- (d) *E1 Tr E2* is partially visible in either *V1* or *V2* or both.

For Security Rules (+13) and (+14):

Let *V1* have IRD-PARTITION-NAME equal to *P1*.  
Let *V2* have IRD-PARTITION-NAME equal to *P2*.

The permission structure for relationships is simpler than for entities. Rather than defining separate ADD, MODIFY, and DELETE permissions for relationships, only UPDATE-PERMISSION is defined. Authorization of UPDATE-PERMISSION for a relationship is defined for two distinct cases:

- (1) The relationship is completely visible in a single IRD-view. This case is covered by Security Rule (+10).
- (2) The relationship is only partially visible in any given IRD-view. This will occur in those cases where *P1* and *P2* are different. It can also occur when different IRD-views have different permissions and/or access-keys. This case is covered by Security Rule (+11).

(+13) *V1* shall authorize UPDATE-PERMISSION for the relationship *E1 Tr E2* if each of the following is satisfied:

- (a) *E1* and *E2* are visible in *V1*.
- (b) At least one of the following is satisfied:

- (i)  $V1$  does not contain an attribute of type EXCLUDE-RELATIONSHIPS which is equal to the assigned-access-name of  $Tr$  and  $V1$  authorizes MODIFY-PERMISSION for either  $E1$  or  $E2$ .
  - (ii)  $V1$  authorizes IRD-ADMINISTRATOR-PERMISSION.
- (+14) Two IRD-views,  $V1$  and  $V2$ , shall authorize UPDATE-PERMISSION for the relationship  $E1$   $Tr$   $E2$  if each of the following is satisfied:
- (a) Both  $V1$  and  $V2$  are related to  $U$ .
  - (b)  $E1$  is visible in  $V1$ .
  - (c)  $E2$  is visible in  $V2$ .
  - (d) At least one of the following is satisfied:
    - (i)  $U$  authorizes IRD-ADMINISTRATOR-PERMISSION.
    - (ii) At least one  $V_i$ ,  $1 \leq i \leq 2$ , authorizes IRD-ADMINISTRATOR-PERMISSION.
    - (iii) At least one  $V_i$ ,  $1 \leq i \leq 2$  does not contain an attribute of type EXCLUDE-RELATIONSHIPS equal to the assigned-access-name of  $Tr$  and at least one  $V_i$ ,  $1 \leq i \leq 2$ , authorizes MODIFY-PERMISSION for  $E_i$ .

### Error and Warning Conditions

- (1) Error E03005: IRDS-user has no corresponding IRD-view. See Security Rule (+1) of this Subsection.
- (2) Error E03006: IRD-view does not identify a valid IRD partition. See Security Rule (+2) of this Subsection.

### 8.2.1 Modifications to IRD Maintenance Commands

In the Subsections which are subordinate to this one, the modifications to IRD commands are given. The meaning of the phrase "authorizes xxx-PERMISSION for a given entity" will be defined within the Security Rules of the appropriate IRD command.

#### 8.2.1.1 Modifications to Add Entity Command

##### Modified Format

```

add-entity-command ::=
    add-word                M1.4.3
    entity-word             M1.4.3
    new-entity-name
    entity-type-clause     M1.6
    [ entity-descriptive-name-declaration-clause ] M1.6.27
    [ quality-indicator-designation-clause ]      M1.6.30
    [ assign-security-clause ] /*New*/           7.8
    [ new-entity-attributes-clause ]             M1.6.22
    ;

```

new-entity-name ::=	
access-name	M1.4.4
null-mark	M1.4.1

For the following Syntax, General, and Security Rules:

Let *E* denote the specified entity.  
Let *T* denote the type of the entity *E*.

### Syntax Rules

- (+1) Assign-security-clause shall not be allowed if the entity-type identified in entity-type-clause is a security entity-type, i.e., ACCESS-CONTROLLER, IRDS-USER, IRD-VIEW, or IRD-SCHEMA-VIEW.
- (+2) If assign-security-clause is specified, the IRD Schema shall contain:
  - (a) The entity-type ACCESS-CONTROLLER in the CONTROLLED IRD Schema life-cycle-phase.
  - (b) The entity-type *T* in the CONTROLLED IRD Schema life-cycle-phase.
  - (c) The relationship-type *T* SECURED-BY ACCESS-CONTROLLER in the CONTROLLED IRD Schema life-cycle-phase.
- (+3) Attributes of type READ-KEY, WRITE-KEY, READ-LOCK, and WRITE-LOCK shall not be specified in new-entity-attributes-clause.

### Modification to General Rules

- (2) If *V* has an IRD-PARTITION-NAME attribute which is not equal to SECURITY, then entities of the following types shall not be added:
  - ACCESS-CONTROLLER
  - IRDS-USER
  - IRD-VIEW
  - IRD-SCHEMA-VIEW
- (3) If *V* has an IRD-PARTITION-NAME attribute equal to SECURITY, then only entities of the following types shall be added:
  - ACCESS-CONTROLLER
  - IRDS-USER
  - IRD-VIEW
  - IRD-SCHEMA-VIEW

### General Rules

- (+1) Assign-security-clause shall not be allowed if *V* has an IRD-PARTITION-NAME attribute equal to SECURITY.



**Security Rules**

- (+1) *V* shall authorize ADD-PERMISSION for entities of type *T*. (In this case it is said that *V* authorizes ADD-PERMISSION for *E*.)
- (+2) Assign-security-clause shall be allowed only if one of the following is satisfied:
  - (a) *V* authorizes ADD-AND-SECURE-PERMISSION for entities of type *T*.
  - (b) *V* authorizes both ADD-ENTITY-PERMISSION and SECURE-ENTITY-PERMISSION for entities of type *T*.

(If either of these is satisfied, then *V* is said to authorize ADD-AND-SECURE-PERMISSION for *E*.)

**Actions**

- (+1) If the command passes validation and with-security-clause is specified, then the specified access-controller shall be assigned to the newly-added entity identified by new-entity-name.

**Error and Warning Conditions**

- (+1) Error E03007: Cannot secure a security entity. See Syntax Rule (+1) and General Rule (+1).
- (+2) Error E03008: Entity-level security cannot be assigned to specified entity. See Syntax Rules (+1) and (+2).
- (+3) Error E03009: System-maintained attributes cannot be specified in IRD maintenance commands. See Syntax Rule (+3).
- (+4) Error E03011: User not authorized to perform specified action(s) by the effective IRD-view. See Security Rules (+1) and (+2).

**8.2.1.2 Modifications to Modify Entity Command****Modified Format**

```

modify-entity-command ::=
    modify-word                M1.4.3
    entity-word                M1.4.3
    existing-entity-name
    [ entity-type-clause ]    M1.6.2
    [ new-version-clause     M1.6.28
    [ exclude-security-clause ] /*New*/ 7.9
    ]
    [ entity-descriptive-name-declaration-clause ] M1.6.27
    [ quality-indicator-designation-clause ]      M1.6.30
    [ modified-entity-attributes-clause ]        M1.6.23
    ;

existing-entity-name ::= access-name                M1.4.4

```

### Syntax Rules

- (+1) Attributes of type READ-KEY, WRITE-KEY, READ-LOCK, and WRITE-LOCK shall not be specified in modified-entity-attributes-clause.

### General Rules

- (+1) If exclude-security-clause is specified for an entity of a type for which entity-level security cannot be defined, a warning shall be issued.

### Security Rules

For the following Security Rules:

Let  $E$  denote the specified meta-entity.  
Let  $T$  denote the type of the meta-entity  $E$ .  
Let  $P1$  denote the IRD Schema life-cycle-phase in which  $E$  exists.

- (+1) If new-version-clause is not specified the following shall be satisfied:
- (a)  $V$  authorizes MODIFY-PERMISSION to entities of type  $T$  in  $P$ .
  - (b)  $P$  equals  $P1$ .
  - (c)  $V$  authorizes MODIFY-PERMISSION for  $E$  (see Security Rule (+9) in Subsection 8.2).
- (+2) If new-version-clause is specified, the following shall be satisfied:
- (a)  $V$  authorizes ADD-PERMISSION for entities of the type  $T$  in  $P$ .
  - (b)  $U$  is related to at least one IRD-view  $V1$  in which  $E$  is visible. An entity which is not visible is treated as if it does not exist.
  - (c) Let  $E'$  denote the entity to be created. Then:
    - (i) Given a relationship of the form  $E R E2$ , there shall be one or two IRD-views  $V [ , Vi ]$  which authorize UPDATE-PERMISSION for the relationship  $E' R E2$ .
    - (ii) Given a relationship of the form  $E1 R E$ , if the corresponding relationship  $E1 R E'$  is to be created, then there shall be one or two IRD-views  $V [ , Vi ]$  which authorize UPDATE-PERMISSION for the relationship  $E1 R E'$ . (Note that not all relationships in which  $E$  participates will generate a corresponding relationship.)

If there is no sequence of IRD-views which authorize the creation of the corresponding relationship, a warning shall be issued and the relationship shall not be created.

Note that the sequence of IRD-views will in each case consist only of  $V$  or  $V$  and one other IRD-view  $Vi$ . Note also that the second IRD-view, if required, can be different for different relationships.

**Actions**

- (+1) If the command passes validation and a new-version-clause is specified, then:
  - (a) If exclude-security-clause is not specified, the new entity shall be assigned the same access-controller(s) as assigned to the entity identified by existing-entity-access-name.
  - (b) Otherwise, the new-entity shall not be assigned any access-controller.

**Error and Warning Conditions**

- (+1) Error E03009: System-maintained attributes cannot be specified in IRD maintenance commands. See Syntax Rule (+1).
- (+2) Warning W03002: Entity-level security not defined for entities of this type. See General Rule (+1).
- (+3) Error E03011: User not authorized to perform specified action(s) by the effective IRD-view. See Security Rule (+1).
- (+4) Error E03004: User not authorized to perform specified action(s). See Security Rule (+2).

**8.2.1.3 Modifications to Delete Entity Command****Modification to General Rule 2**

- (2) If with-relationships-clause is not specified, each entity to be deleted shall not participate in any relationship except for those relationships in which the other entity is of type ACCESS-CONTROLLER.

**Security Rules**

For the following rules:

Let *E* denote an entity to be deleted.

- (+1) *V* shall authorize DELETE-PERMISSION for each *E* (see Security Rule (+9) of Subsection 8.2).
- (+2) If with-relationships-clause is specified, then *V* shall authorize DELETE-PERMISSION for each relationship in which *E* participates. If this is not the case, no action shall take place.

**Actions**

- (+1) If the entity is secured, then all relationships to entities of type ACCESS-CONTROLLER shall be deleted.

**Error and Warning Conditions**

- (+1) Error E03011: User not authorized to perform specified action(s) by the effective IRD-view. See Security Rule (+1).

- (+2) Error E03004: User not authorized to perform specified action(s). See Security Rule (+2).

#### 8.2.1.4 Modifications to Add Relationship Command

##### General Rules

- (+1) This command shall not be operable on any relationship in which an entity of type ACCESS-CONTROLLER participates.

##### Security Rules

For the following rules:

Let  $Tr$  denote the type of the specified relationship.  
Let  $E1$  denote the first entity in the relationship.  
Let  $E2$  denote the second entity in the relationship.  
Let  $T1$  and  $T2$  denote the types of  $E1$  and  $E2$ , respectively.  
Let  $P1$  denote the IRD-partition in which  $E1$  exists.  
Let  $P2$  denote the IRD-partition in which  $E2$  exists.

- (+1) If  $P1 = P2$ , one of the following shall be satisfied:
- (a) Each of the following is satisfied:
    - (i)  $P = P1$ .
    - (ii)  $V$  authorizes UPDATE-PERMISSION for  $E1 Tr E2$ , as stated in Security Rule (+13) of Subsection 8.2.
  - (b) Each of the following is satisfied:
    - (i)  $P = P1$ .
    - (ii) There exists  $V2$  such that  $V$  and  $V2$  authorize UPDATE-PERMISSION for  $E1 Tr E2$  as stated in Security Rule (+14) of Subsection 8.2.
- (+2) If  $P1$  is not equal to  $P2$ , then each of the following shall be satisfied:
- (a)  $P = P1$ .
  - (b) There exists  $V2$  such that  $V$  and  $V2$  authorize UPDATE-PERMISSION for  $E1 Tr E2$  as stated in Security Rule (+14) of Subsection 8.2.

##### Error and Warning Conditions

- (+1) Error E03012: Relationships which implement entity-level security cannot be maintained by normal maintenance commands. See General Rule (+1).
- (+2) Error E03004: User not authorized to perform specified action(s). See Security Rules (+1) and (+2).

### 8.2.1.5 Modifications to Modify Relationship Command

#### General Rules

- (+1) This command shall not be operable on any relationship in which an entity of type ACCESS-CONTROLLER participates.

#### Security Rules

For the following rules:

Let  $Tr$  denote the type of the specified relationship.  
 Let  $E1$  denote the first entity in the relationship.  
 Let  $E2$  denote the second entity in the relationship.  
 Let  $T1$  and  $T2$  denote the types of  $E1$  and  $E2$ , respectively.  
 Let  $P1$  denote the IRD-partition in which  $E1$  exists.  
 Let  $P2$  denote the IRD-partition in which  $E2$  exists.

- (+1) If  $P1 = P2$ , one of the following shall be satisfied:
- (a) Each of the following is satisfied:
    - (i)  $P = P1$ .
    - (ii)  $V$  authorizes UPDATE-PERMISSION for  $E1 Tr E2$ , as stated in Security Rule (+13) of Subsection 8.2.
  - (b) Each of the following is satisfied:
    - (i)  $P = P1$ .
    - (ii) There exists  $V2$  such that  $V$  and  $V2$  authorize UPDATE-PERMISSION for  $E1 Tr E2$  as stated in Security Rule (+14) of Subsection 8.2.
- (+2) If  $P1$  is not equal to  $P2$ , then each of the following shall be satisfied:
- (a)  $P = P1$ .
  - (b) There exists  $V2$  such that  $V$  and  $V2$  authorize UPDATE-PERMISSION for  $E1 Tr E2$  as stated in Security Rule (+14) of Subsection 8.2.

#### Error and Warning Conditions

- (+1) Error E03012: Relationships which implement entity-level security cannot be maintained by normal maintenance commands. See General Rule (+1).
- (+2) Error E03004: User not authorized to perform specified action(s). See Security Rules (+1) and (+2).

### 8.2.1.6 Modifications to Delete Relationship Command

#### General Rules

- (+1) This command shall not be operable on any relationship in which an entity of type ACCESS-CONTROLLER participates.

#### Security Rules

For the following rules:

Let  $Tr$  denote the type of the specified relationship.  
Let  $E1$  denote the first entity in the relationship.  
Let  $E2$  denote the second entity in the relationship.  
Let  $T1$  and  $T2$  denote the types of  $E1$  and  $E2$ , respectively.  
Let  $P1$  denote the IRD-partition in which  $E1$  exists.  
Let  $P2$  denote the IRD-partition in which  $E2$  exists.

- (+1) If  $P1 = P2$ , one of the following shall be satisfied:
- (a) Each of the following is satisfied:
    - (i)  $P = P1$ .
    - (ii)  $V$  authorizes UPDATE-PERMISSION for  $E1 Tr E2$ , as stated in Security Rule (+13) of Subsection 8.2.
  - (b) Each of the following is satisfied:
    - (i)  $P = P1$ .
    - (ii) There exists  $V2$  such that  $V$  and  $V2$  authorize UPDATE-PERMISSION for  $E1 Tr E2$  as stated in Security Rule (+14) of Subsection 8.2.
- (+2) If  $P1$  is not equal to  $P2$ , then each of the following shall be satisfied:
- (a)  $P = P1$ .
  - (b) There exists  $V2$  such that  $V$  and  $V2$  authorize UPDATE-PERMISSION for  $E1 Tr E2$  as stated in Subsection 8.2., Security Rule (+14).

#### Error and Warning Conditions

- (+1) Error E03012: Relationships which implement entity-level security cannot be maintained by normal maintenance commands. See General Rule (+1).
- + (2) Error E03004: User not authorized to perform specified action(s). See Security Rules (+1) and (+2).

### 8.2.1.7 Modifications to Modify Entity Access-Name Command

#### Security Rules

- (+1)  $U$  shall be authorized IRD-RENAME-PERMISSION.

**Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action(s). See Security Rule (+1).

**8.2.1.8 Modifications to Modify Entity Descriptive-Name Command****Security Rules**

- (+1) *U* shall be authorized IRD-RENAME-PERMISSION.

**Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action(s). See Security Rule (+1).

**8.2.1.9 Modifications to Modify Entity Life-Cycle-Phase Command****Modified Format**

```

modify-entity-life-cycle-phase-command ::=
    modify-word           M1.4.3
    entity-word           M1.4.3
    life-cycle-phase-word M1.4.3
    [ for-word ]         M1.4.3
    entity-name-list     M1.5.2.1.9
    from-word            M1.4.3
    current-life-cycle-phase M1.5.2.1.9
    to-word              M1.4.3
    new-life-cycle-phase M1.5.2.1.9
    [ new-version-clause M1.6.28
    [ exclude-security-clause ] /*New*/ 7.9
    ]
    [ assign-security-clause ] /*New*/ 7.8
    ;

```

**Security Rules**

For the following Security Rules:

Let  $E1 \dots En$  denote entities specified in the command.  
 Let  $T1 \dots Tn$  denote the corresponding type for  $E1 \dots En$ , respectively.  
 Let  $P1$  denote current-life-cycle-phase.  
 Let  $P2$  denote new-life-cycle-phase.  
 Let  $V1 \dots Vm$  denote IRD-views related to  $U$ .

- (+1) If new-version-clause is not specified then each of the following shall be satisfied:
- (a)  $V$  authorizes ADD-ENTITY-PERMISSION for entities of each type  $Ti$  in  $P1$ .
  - (b) For each  $Ei$ , there exists an IRD-view  $Vj$  related to  $U$  with the IRD-PARTITION-NAME attribute equal to 1 such that  $Ei$  is visible in  $Vj$  and  $Vi$  authorizes MODIFY-PHASE-PERMISSION for entities of type  $Ti$ .

- (c) If  $E_i$  is secured, then  $V$  has a write-key corresponding to a write-lock on at least one access-controller assigned to  $E_i$ .
- (+2) If new-version-clause is specified then:
  - (a)  $V$  shall be authorized to ADD-PERMISSION for entities of each  $T_i$  in  $P_2$ .
  - (b) Each  $E_i$  shall be visible in some  $V_j$ ,  $1 \leq j \leq m$ .
- (+3) If assign-security-clause is specified,  $V$  shall authorize either ADD-AND-SECURE-PERMISSION or SECURE-ENTITY-PERMISSION for entities of each type  $T_i$ .
- (+4) If exclude-security-clause is specified,  $V$  shall authorize SECURE-ENTITY-PERMISSION for entities of type  $T_i$ .

**Actions**

- (+1) If the command passes validation and a new-version-clause is specified, then for each entity in entity-name-list which passes all additional constraints:
  - (a) If exclude-security-clause has been specified, the new entity shall not be assigned any of the access-controllers which are assigned to the corresponding entity in entity-name-list. Otherwise, the new entity shall be created with the same access-controllers assigned to it as the corresponding entity in entity-name-list.
  - (b) If assign-security-clause has been specified, the new entity shall be assigned the specified access-controller.
- (+2) If the command passes validation, new-version-clause is not specified, and assign-security-clause is specified, the specified access-controller shall be assigned to each entity identified by new-entity-access-name which passes all additional constraints.

**Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action(s). See Security Rules (+1) and (+2).
- (+2) Error E03011: User not authorized to perform specified action(s) by the effective IRD-view. See Security Rules (+3) and (+4).

**8.2.1.10 Modifications to Copy Entity Command**

**Modified Format**

```

copy-entity-command ::=
    copy-word                M1.4.3
    entity-word              M1.4.3
    existing-entity-access-name M1.5.2.1.10
    [ with-relationships-clause ] M1.6.32
    to-word                  M1.4.3
    new-entity-access-name
    [ entity-descriptive-name-declaration-clause ] M1.6.27
    [ quality-indicator-designation-clause ] M1.6.30
    
```



[ exclude-security-clause ] /*New*/	7.9
[ assign-security-clause ] /*New*/	7.8
;	

new-entity-access-name ::=	
access-name	M1.4.4
null-mark	M1.4.1
new-version-clause	M1.6.28

### General Rules

- (+1) The entity-type of the entity identified by existing-entity-access-name shall not be ACCESS-CONTROLLER.
- (+2) With-relationships-clause applies only to relationships in which an entity of type ACCESS-CONTROLLER does not participate. Relationships in which an ACCESS-CONTROLLER entity participates shall be created if and only if exclude-security-clause is not specified.
- (+3) If the effective IRD-view is associated with the IRD partition named SECURITY, assign-security-clause shall not be allowed.

### Security Rules

For the following Security Rules:

Let  $E$  denote the entity identified by existing-entity-access-name.  
 Let  $E'$  denote the new entity to be created.  
 Let  $R$  denote a relationship-type.  
 Let  $E1$  denote an entity such that  $E1 R E$  exists.  
 Let  $E2$  denote an entity such that  $E R E2$  exists.  
 Let  $T$  denote the type corresponding to  $E$ .

- (+1)  $V$  shall authorize ADD-PERMISSION to meta-entities of type  $T$ .
- (+2)  $U$  shall be related to an IRD-view  $V1$  in which  $E$  is visible.
- (+3) If assign-security-clause is specified,  $V$  shall authorize either ADD-AND-SECURE-PERMISSION or SECURE-ENTITY-PERMISSION for entities of type  $T$ .
- (+4) If exclude-security-clause is specified,  $V$  shall authorize SECURE-ENTITY-PERMISSION for entities of type  $T$ .
- (+5) If with-relationships-clause is specified, then:
  - (a) Given a relationship of the form  $E R E2$ , there shall be one or two IRD-views  $V [ , Vi ]$  which authorize UPDATE-PERMISSION for the relationship  $E' R E2$ .
  - (b) Given a relationship of the form  $E1 R E$ , if the corresponding relationship  $E1 R E'$  is to be created, then there shall be one or two IRD-views  $V [ , Vi ]$  which authorize UPDATE-PERMISSION for the relationship  $E1 R E'$ . (Note that in this case, not all relationships in which  $E$  participates will generate a corresponding new relationship.)

If there is no sequence of IRD-views which authorize the creation of the corresponding relationship, a warning shall be issued and the relationship shall not be created.

Note that the sequence of IRD-views will in each case consist only of  $V$  or  $V$  and one other IRD-view  $V_i$ . Note also that the second IRD-view, if required, can be different for different generated relationships.

### Actions

- (+1) The access-controllers assigned to the entity identified by existing-entity-access-name shall also be assigned to the new entity identified by new-entity-access-name unless exclude-security-clause is specified.
- (+2) If assign-security-clause is specified, the specified access-controller shall be assigned to the entity identified by new-entity-access-name.

### Error and Warning Conditions

- (+1) Error E03013: An access-controller cannot be copied. General Rule (+1) has been violated.
- (+2) Error E03007: Cannot Secure a security entity. See General Rule (+3).
- (+3) Error E03011: User not authorized to perform specified action(s) by the effective IRD-view. See Security Rules (+1), (+3), (+4).
- (+4) Error E03004: User not authorized to perform specified action(s). See Security Rules (+2) and (+5).

### 8.2.2 Modifications to IRD Output Functions

The following changes apply to all IRD output commands. Modifications to the output-impact-of-change and output-syntax functions are also specified in subordinate Subsections. The general-output function is not affected.

### Security Rules

- (+1) If specified, using-IRD-views-clause shall restrict entity and relationship visibility only to those IRD-views identified within the command-clause.
- (+2) If a relationship:
  - (a) Is of a type which is visible in the IRD-view or views being used by this command,
  - (b) Relationships of this type are to be shown by the command, and
  - (c) Is only partially visible in those IRD-views, then the relationship and all its attributes shall be displayed, but the access-name of the entity not visible to the specified IRD-views shall be hidden. In place of the name components, strings of fill characters (X) shall be displayed. The length of each string shall be the maximum length of the component of the access-name.

- (+3) If an entity satisfies the specified selection criteria, but:
- (a) It is of an entity-type which is not visible within the IRD-views used by this command, and
  - (b) It is not explicitly specified within the selection criteria,
- then it shall be bypassed.
- (+4) If a show-clause (qualified or unqualified) specifies all relationships, only those relationships which are (partially or completely) visible within the IRD-views used by the command shall be selected and displayed. No error or warning condition shall exist.
- (+5) Unless otherwise specified, a predefined display is specified in show-option, all information on the display shall be visible within the IRD-views used by the command. In particular, if any entity-type, attribute-type, attribute-group-type, or relationship-type which appears in the display is not visible within the IRD-views used by the command, an error condition shall exist.
- (+6) If a command-clause explicitly specifies an entity-type, relationship-type, attribute-group-type or attribute-type which is not visible within the IRD-views used by the command, an error condition shall exist. This applies to subclauses of both the show-clause and the entity-selection-criteria-clause.
- (+7) The selection-criteria "WHERE RELATIONSHIPS EXIST" and "WHERE NO RELATIONSHIPS EXIST" shall be evaluated without regard to whether or not those relationships in which the tested entity participates are visible or not.
- (+8) The relationships which implement entity-level security shall not be used in evaluating the selection criteria "WHERE RELATIONSHIPS EXIST" and "WHERE NO RELATIONSHIPS EXIST".

### Error and Warning Conditions

- (+1) Error E03014: IRD Schema descriptor not visible to user. See Security Rules (+5) and (+6).

#### 8.2.2.1 Modifications to Output Impact-of-Change Command

##### Security Rules

For the following Security Rules:

Let  $E(s)$  denote an entity specified in the output-impact-of-change command.  
Let  $E'$  denote an entity potentially impacted by a change in  $E(s)$ .

- (+1) If  $E'$  is not visible within any of the IRD-views used in the command:
- (a) The impact on  $E'$  shall not be expanded.
  - (b) The name of  $E'$  shall be hidden by fill characters (X). Each component of its access-name (and descriptive-name if it is also to be displayed) shall be replaced by fill characters to the maximum length of the access-name component.

- (c) No additional information about *E'* shall be displayed, regardless of the show-clauses specified in impact-show-options.
- (d) A warning condition shall exist.
- (e) An implementation may suppress the display of hidden entity names.

### **Error and Warning Conditions**

- (+1) Warning W03003: Impacted entity not visible. See Security Rule (1)(d).

### **8.2.2.2 Modifications to Output Syntax Command**

#### **Security Rules**

- (+1) If all relationships are requested, only relationships of types visible to the user shall be displayed. This overrides Security Rule (+5) of IRD Output Commands.

### **8.3 Modifications to General Commands**

Modifications to General Commands are identified in subordinate Subsections. Only the affected commands are identified. These commands are as follows:

```
create-IRD-command
export-IRD-command
check-IRD-schema-compatibility-command
import-IRD-command
help-command
enter-panel-dialogue-command
```

The following notation is used as required in the modifications to each of the above commands.

Let *U* denote the effective IRDS-user.  
Let *Vs* denote the effective IRD-schema-view.  
Let *Vd* denote the effective IRD-view.

#### **8.3.1 Modifications to Create IRD Command**

##### **Security Rules**

- (+1) Authority to issue the create-IRD-command will have to be established outside the IRDS, since this command cannot be issued before there is an IRDS in existence.
- (+2) If IRD-schema-source-clause specifies another IRDS, the user shall establish authorization to read the IRD Schema in that IRDS. The mechanism for establishing such authorization shall be implementor-defined. Implementor-defined-clauses may be used for this purpose.
- (+3) If IRD-schema-source-clause specifies another IRDS, the user shall be authorized READ-PERMISSION for all meta-entities and meta-relationships in the CONTROLLED IRD Schema life-cycle-phase of the IRD Schema for the specified IRDS.

**Error and Warning Conditions**

- (+1) Error E03015: User does not have adequate authorization to source IRD Schema. See Security Rule (+3).

**8.3.2 Modifications to Export IRD Command****Security Rules**

- (+1) Vs shall authorize READ-PERMISSION for all meta-entities and meta-relationships in the CONTROLLED IRD Schema life-cycle-phase.
- (+2) If using-list-clause is specified, no further security checking shall be required.
- (+3) If entity-selection-criteria-clause is specified, then:
  - (a) Only entities which are visible to an IRDS-user shall be extracted. The set of entities which are visible to an IRDS-user in this command shall be those which are visible by the IRD-views specified in the using-IRD-views-clause. If no using-IRD-views-clause is specified, then only those entities visible in the effective IRD-view shall be extracted. Any entity which meets the selection-criteria but is not visible in the IRD-views used by the command shall be ignored.
  - (b) Only completely visible relationships shall be extracted.
  - (c) The selection-criteria "WHERE RELATIONSHIPS EXIST" and "WHERE NO RELATIONSHIPS EXIST" shall be evaluated without regard to whether or not the IRDS-user is authorized READ-PERMISSION for all relationships in which the tested entity participates.
  - (d) The relationships which implement entity-level security shall not be used in evaluating the selection criteria "WHERE RELATIONSHIPS EXIST" and "WHERE NO RELATIONSHIPS EXIST".

**Error and Warning Conditions**

- (+1) Error E03015: User does not have adequate authorization to source IRD Schema. See Security Rule (+1).

**8.3.3 Modifications to Check IRD Schema Compatibility Command****Security Rules**

- (+1) Vs shall authorize READ PERMISSION for all meta-entities and meta-relationships in the CONTROLLED IRD Schema life-cycle-phase for each IRDS which is in use.
- (+2) If in-IRD-clause is specified in the other-IRD-schema-clause, the user shall establish authorization to read the IRD Schema. The mechanism for establishing such authorization shall be implementor-defined. Implementor-defined-clauses may be used for this purpose.

### **Error and Warning Conditions**

- (+1) Error E03016: User does not have adequate authorization to current or specified IRD Schema. See Security Rule (+1).

### **8.3.4 Modifications to Import IRD Command**

#### **Security Rules**

- (+1) *Vd* shall authorize ADD-PERMISSION for each entity in the associated IRD-partition.

### **Error and Warning Conditions**

- (+1) Error E03011: User not authorized to perform the specified action(s) by the effective IRD-view.

### **8.3.5 Modifications to Help Command**

#### **Security Rules**

- (+1) The commands displayed by the help-command shall be restricted to those which a user is authorized.
- (+2) In the help area of the panel interface, only information about actions and other panels which the user is authorized to use shall be displayed.

### **8.3.6 Modifications to Enter Panel Dialogue Command**

#### **Security Rules**

- (+1) *U* shall be authorized PANEL-PERMISSION.
- (+2) The panels which a user can specify, either by name or by command-imperative, shall be restricted to those for which the user is authorized.

### **Error and Warning Conditions**

- (+1) Error E03004: User not authorized to perform specified action(s). See Security Rules (+1) and (+2).

## **8.4 Modifications to Panel Trees**

All panel trees in Module 1 shall accommodate all the changes dictated by the new and modified rules specified in Subsections 8.1 through 8.3 inclusive. All new error and warning messages shall also be accommodated.

The only panel trees which have additional changes are those which correspond to commands in which the syntax has been modified. These panel trees are as follows:

- (1) add-entity-panel-tree
- (2) modify-entity-panel-tree
- (3) modify-entity-life-cycle-phase-panel-tree
- (4) copy-entity-panel-tree

#### 8.4.1 Modifications to Add Entity Panel Tree

In addition to the Data Area contents defined in Module 1, the Data Area shall prompt for an access-controller-name where appropriate. This change corresponds to the assign-security-clause which has been added to the add-entity-command.

#### 8.4.2 Modifications to Modify Entity Panel Tree

In addition to the Data Area contents defined in Module 1, the Data Area shall include the following: When a new version-identifier of the entity specified is entered, a prompt shall be added where appropriate to determine if the access-controllers assigned to the existing entity are to be assigned to the new entity or not. This change corresponds to the exclude-security-clause which has been added to the modify-entity-command.

#### 8.4.3 Modifications to Modify Entity Life-Cycle-Phase Panel Tree

In addition to the Data Area contents defined in Module 1, the Data Area shall include the following:

- (1) When a new version-identifier of the entity specified is entered, a prompt shall be added where appropriate to determine if the access-controllers assigned to the existing entity are to be assigned to the new entity or not. This change corresponds to the exclude-security-clause which has been added to the corresponding command.
- (2) The Data Area shall prompt for an access-controller-name where appropriate. This change corresponds to the assign-security-clause which has been added to the modify-entity-life-cycle-phase-command.

#### 8.4.4 Modifications to Copy Entity Panel Tree

In addition to the Data Area contents defined in Module 1, the Data Area shall include the following:

- (1) A prompt shall be added where appropriate to determine if the access-controllers assigned to the existing entity shall also be assigned to the new entity or not. This change corresponds to the exclude-security-clause which has been added to the copy-entity-command.
- (2) The Data Area shall prompt for an access-controller-name where appropriate. This change corresponds to the assign-security-clause which has been added to the copy-entity-command.

### 9 Modifications to the IRD Schema

This section takes the form of IRD Schema commands. These IRD Schema commands identify the changes to the IRD Schema identified in Module 1.

The commands are divided into the following subsections:

- (1) New entity-types
- (2) New relationship-types

- (3) New attribute-types
- (4) New attribute-group-types
- (5) Changes to existing entity-types
- (6) Associations of entity-types with new attribute-types
- (7) Associations of entity-types with new attribute-group-types
- (8) Reserved-names

Although the form of IRD Schema commands is used to describe the required changes, an implementation of the IRDS need not use this approach for installation of the IRDS Security Module. Likewise, these commands do not specify the movement of the meta-entities into the CONTROLLED IRD Schema life-cycle-phase which would also be required to support this Module.

Note that PURPOSE meta-attributes are used as a documentation technique for this Module. In an actual implementation, however, this standard does not require the PURPOSE meta-attributes. An implementation may use the PURPOSE attributes as specified below if it so desires. Alternatively, it may either ignore specifying the PURPOSE meta-attributes, or use its own wording. The wording specified for these meta-attributes may be regarded as a suggestion for how the various types required by this Module should be documented in the IRD Schema.

Comments are also used in these command streams. These comments identify groups of related commands and discuss installation options.

### 9.1 New Entity-Types

/\* The following IRD Schema command defines the entity-type which supports the IRDS entity-level security facility \*/

```

ADD META-ENTITY ACCESS-CONTROLLER
    META-ENTITY-TYPE = ENTITY-TYPE
    WITH META-ATTRIBUTES
        ENTITY-CLASS = SECURITY
        ORIGIN = X3.138-1988 (Ch.3)
        PURPOSE =
    "This entity-type is required for the entity-level security
    facilities defined in Module 3 of the standard IRDS."
    ;
    
```

### 9.2 New Relationship-Types

An installation can decide to implement entity-level security selectively for a specified set of entity-types. For each such entity-type whose assigned-access-name is denoted here by "\$\$X", the IRD Schema commands given below shall establish entity-level security for entities of that type. Note here that \$\$X shall not identify any security entity-type (ACCESS-CONTROLLER, IRDS-USER, IRD-VIEW, IRD-SCHEMA-VIEW). Note also that in order to take effect, the entity-type, relationship-type and relationship-class-type meta-entities all shall be in the CONTROLLED IRD-schema-life-cycle-phase.

```

ADD META-ENTITY SECURED-BY
    META-ENTITY-TYPE = RELATIONSHIP-CLASS-TYPE
    WITH META-ATTRIBUTES
        ORIGIN = X3.138-1988 (Ch.3)
        INVERSE-NAME = SECURES
    
```



PURPOSE =  
 "This relationship-class-type identifies all relationship-types used to support IRDS entity-level security."  
 ;

ADD META-ENTITY \$\$X-SECURED-BY-ACCESS-CONTROLLER  
 META-ENTITY-TYPE = RELATIONSHIP-TYPE  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 INVERSE-NAME = ACCESS-CONTROLLER-SECURES-\$\$X  
 PURPOSE =  
 "The existence of this relationship-type enables any entity of type \$\$X to be secured. That is, the add-security, modify-security, and delete-security command can operate on entities of type \$\$X provided the IRDS-user who submits the command has the appropriate authorization."  
 ;

ADD META-RELATIONSHIP  
 \$\$X-SECURED-BY-ACCESS-CONTROLLER MEMBER-OF SECURED-BY  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 ;

ADD META-RELATIONSHIP  
 \$\$X-SECURED-BY-ACCESS-CONTROLLER CONNECTS \$\$X  
 POSITION = 1  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 ;

ADD META-RELATIONSHIP  
 \$\$X-SECURED-BY-ACCESS-CONTROLLER CONNECTS ACCESS-CONTROLLER  
 POSITION = 2  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 ;

### 9.3 New Attribute-Types

/\* The following IRD Schema commands define the attribute-types which support the IRDS global security facilities. \*/

ADD META-ENTITY ADD-PERMISSION  
 META-ENTITY-TYPE = ATTRIBUTE-TYPE  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 FORMAT = STRING  
 MINIMUM-ATTRIBUTE-LENGTH = 1  
 MAXIMUM-ATTRIBUTE-LENGTH = 1  
 PURPOSE =  
 "This attribute-type is associated with two attribute-group-types: IRD-PERMISSIONS and IRD-SCHEMA-PERMISSIONS. In an IRD-PERMISSIONS

attribute-group, a YES in the attribute of this type indicates that the user can add entities of the type identified in the corresponding ENTITY-TYPE-NAME attribute.

Similarly, in an IRD-SCHEMA-PERMISSIONS attribute group, a YES in the attribute of this type indicates that the user can add meta-entities of the type identified in the corresponding META-ENTITY-TYPE-NAME attribute."

;

ADD META-ENTITY COMMAND-LANGUAGE-PERMISSION

META-ENTITY-TYPE = ATTRIBUTE-TYPE

WITH META-ATTRIBUTES

ORIGIN = X3.138-1988 (Ch.3)

FORMAT = STRING

MINIMUM-ATTRIBUTE-LENGTH = 1

MAXIMUM-ATTRIBUTE-LENGTH = 1

PURPOSE =

"This attribute-type is associated with the entity-type of IRDS-USER. For an given IRDS-user entity, a YES in the attribute of this type indicates that the user can use the command language interface. Attributes of this type are meaningful only if the implementation provides a command language interface."

;

ADD META-ENTITY DELETE-PERMISSION

META-ENTITY-TYPE = ATTRIBUTE-TYPE

WITH META-ATTRIBUTES

ORIGIN = X3.138-1988 (Ch.3)

FORMAT = STRING

MINIMUM-ATTRIBUTE-LENGTH = 1

MAXIMUM-ATTRIBUTE-LENGTH = 1

PURPOSE =

"This attribute-type is associated with two attribute-group-types: IRD-PERMISSIONS and IRD-SCHEMA-PERMISSIONS. In an IRD-PERMISSIONS attribute-group, a YES in the attribute of this type indicates that the user can delete entities of the type identified in the corresponding ENTITY-TYPE-NAME attribute.

Similarly, in an IRD-SCHEMA-PERMISSIONS attribute group, a YES in the attribute of this type indicates that the user can delete meta-entities of the type identified in the corresponding META-ENTITY-TYPE-NAME attribute."

;

ADD META-ENTITY IRD-ADMINISTRATOR-PERMISSION

META-ENTITY-TYPE = ATTRIBUTE-TYPE

WITH META-ATTRIBUTES

ORIGIN = X3.138-1988 (Ch.3)

FORMAT = STRING

MINIMUM-ATTRIBUTE-LENGTH = 1

MAXIMUM-ATTRIBUTE-LENGTH = 1

PURPOSE =

"This attribute-type is associated with two entity-types: IRDS-USER

and IRD-VIEW. When an attribute of this type equals YES in an IRD-view, it identifies that all IRD permissions are granted for any related IRDS-user in the IRD partition identified by the corresponding IRD-PARTITION-NAME attribute. This overrides whatever is specified in the IRD-PERMISSIONS attribute-groups. Furthermore, it also grants the user all permissions to any secured entity within the corresponding IRD partition.

Whenever an attribute of this type equal YES in an entity of type IRDS-USER, the IRDS-user is granted all privileges in all IRD partitions. Furthermore, the IRDS-user is also granted IRD-RENAME-PERMISSION, and where applicable, both COMMAND-LANGUAGE-PERMISSION and PANEL-PERMISSION. Finally, an IRDS-user, to which this permission has been granted has all permissions to any secured entity in the IRD."

;

```
ADD META-ENTITY IRD-RENAME-PERMISSION
  META-ENTITY-TYPE = ATTRIBUTE-TYPE
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    FORMAT = STRING
    MINIMUM-ATTRIBUTE-LENGTH = 1
    MAXIMUM-ATTRIBUTE-LENGTH = 1
    PURPOSE =
```

"This attribute-type is associated with the IRDS-USER entity-type. When an attribute of this type equals YES, the IRDS-user is authorized to use the modify-entity-access-name command and modify-entity-descriptive-name command.

This permission is independent of IRDS entity security, and implies READ-PERMISSION to all secured entities."

;

```
ADD META-ENTITY ENTITY-TYPE-NAME
  META-ENTITY-TYPE = ATTRIBUTE-TYPE
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    FORMAT = STRING
    MINIMUM-ATTRIBUTE-LENGTH = 1
    MAXIMUM-ATTRIBUTE-LENGTH = 32
    PURPOSE =
```

"This attribute-type is the key attribute-type for the attribute-group-type IRD-PERMISSIONS. The meaningful values are 'ALL' and the assigned-access-name of entity-type meta-entities in the CONTROLLED IRD Schema life-cycle-phase. Within any given IRD-PERMISSIONS attribute-group, those non-key attributes with a value of YES identify the permissions authorized to entities of the type identified by the key attribute. If the ENTITY-TYPE-NAME attribute equals ALL, this is taken to mean all entity-types defined by meta-entities in the CONTROLLED IRD Schema life-cycle-phase."

;

ADD META-ENTITY EXCLUDE-META-RELATIONSHIPS

META-ENTITY-TYPE = ATTRIBUTE-TYPE  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 FORMAT = STRING  
 MINIMUM-ATTRIBUTE-LENGTH = 1  
 MAXIMUM-ATTRIBUTE-LENGTH = 72  
 PURPOSE =

"This attribute-type is associated with the entity-type of IRD-SCHEMA-VIEW. If a given IRD-schema-view authorizes READ-PERMISSION for a meta-entity-type, A, READ-PERMISSION is also granted for all meta-relationship-types in which A participates. If any such meta-relationship-type is specified in an attribute of this type, then READ-PERMISSION for that meta-relationship-type is denied. This also excludes the possibility of maintaining meta-relationships of such types."

;

ADD META-ENTITY EXCLUDE-RELATIONSHIPS

META-ENTITY-TYPE = ATTRIBUTE-TYPE  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 FORMAT = STRING  
 MINIMUM-ATTRIBUTE-LENGTH = 1  
 MAXIMUM-ATTRIBUTE-LENGTH = 72  
 PURPOSE =

"This attribute-type is associated with the entity-type of IRD-VIEW. If a given IRD-view authorizes READ-PERMISSION for an entity-type, A, READ-PERMISSION is also granted for all relationship-types in which A participates. If the assigned-access-name of any such relationship-type meta-entity in the CONTROLLED IRD Schema life-cycle-phase is specified in an attribute of this type, then READ-PERMISSION for that relationship-type is denied. This also excludes the possibility of maintaining relationships of such types."

;

ADD META-ENTITY IRD-SCHEMA-ADMINISTRATOR-PERMISSION

META-ENTITY-TYPE = ATTRIBUTE-TYPE  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)  
 FORMAT = STRING  
 MINIMUM-ATTRIBUTE-LENGTH = 1  
 MAXIMUM-ATTRIBUTE-LENGTH = 1  
 PURPOSE =

"This attribute-type is associated with the entity-type of IRDS-USER. For a given IRDS-user entity, a YES in the attribute of this type indicates that the user has unrestricted use of all IRD Schema commands in all IRD Schema life-cycle-phases."

;

ADD META-ENTITY IRD-SCHEMA-RENAME-PERMISSION

META-ENTITY-TYPE = ATTRIBUTE-TYPE  
 WITH META-ATTRIBUTES  
 ORIGIN = X3.138-1988 (Ch.3)

```

FORMAT = STRING
MINIMUM-ATTRIBUTE-LENGTH = 1
MAXIMUM-ATTRIBUTE-LENGTH = 1
PURPOSE =

```

"This attribute-type is associated with the IRDS-USER entity-type. When an attribute of this type equals YES, the IRDS-user is authorized to use the modify meta-entity access-name-command and modify- meta-entity-descriptive-name-command."

```

;
```

```

ADD META-ENTITY META-ENTITY-TYPE-NAME
META-ENTITY-TYPE = ATTRIBUTE-TYPE
WITH META-ATTRIBUTES
ORIGIN = X3.138-1988 (Ch.3)
FORMAT = STRING
MINIMUM-ATTRIBUTE-LENGTH = 1
MAXIMUM-ATTRIBUTE-LENGTH = 32
PURPOSE =

```

"This attribute-type is the key attribute-type for the attribute-group-type IRD-SCHEMA-PERMISSIONS. The meaningful values are 'ALL' and any meta-entity-type. Within any given IRD-SCHEMA-PERMISSIONS attribute-group, those non-key attributes with a value of YES identify the permissions authorized to meta-entities of the type identified by the key attribute. If the META-ENTITY-TYPE-NAME attribute equals ALL, this is taken to mean all meta-entity-types."

```

;
```

```

ADD META-ENTITY MODIFY-PERMISSION
META-ENTITY-TYPE = ATTRIBUTE-TYPE
WITH META-ATTRIBUTES
ORIGIN = X3.138-1988 (Ch.3)
FORMAT = STRING
MINIMUM-ATTRIBUTE-LENGTH = 1
MAXIMUM-ATTRIBUTE-LENGTH = 1
PURPOSE =

```

"This attribute-type is associated with two attribute-group-types: IRD-PERMISSIONS and IRD-SCHEMA-PERMISSIONS. In an IRD-PERMISSIONS attribute-group, a YES in the attribute of this type indicates that the user can modify entities of the type identified in the corresponding ENTITY-TYPE-NAME attribute.

Similarly, in an IRD-SCHEMA-PERMISSIONS attribute group, a YES in the attribute of this type indicates that the user can modify meta-entities of the type identified in the corresponding META-ENTITY-TYPE-NAME attribute."

```

;
```

```

ADD META-ENTITY MODIFY-PHASE-PERMISSION
META-ENTITY-TYPE = ATTRIBUTE-TYPE
WITH META-ATTRIBUTES
ORIGIN = X3.138-1988 (Ch.3)
FORMAT = STRING
MINIMUM-ATTRIBUTE-LENGTH = 1
MAXIMUM-ATTRIBUTE-LENGTH = 1

```

PURPOSE =

"This attribute-type is associated with the attribute-group-type IRD-PERMISSIONS. In an IRD-PERMISSIONS attribute-group, a YES in the attribute of this type indicates that the user can move entities of the type identified in the corresponding ENTITY-TYPE-NAME attribute from the IRD life-cycle-phase corresponding to the IRD-view in which the attribute-group exists.

The following conditions identify the minimal permissions that shall be satisfied for an IRDS-user U to be authorized to modify the life-cycle-phase of an entity of type T from a phase P1 to a phase P2:

- (1) U is related to at least two IRD-view entities, V1 and V2.
- (2) The IRD-PARTITION-NAME attribute of V1 equals P1.
- (3) The IRD-PARTITION-NAME attribute of V2 equals P2.
- (4) V1 has an IRD-PERMISSIONS attribute-group with MODIFY-PHASE-PERMISSION equal to YES for entities of type T.
- (5) V2 has an IRD-PERMISSIONS attribute-group with ADD-PERMISSION equal to YES for entities of type T.

Combinations of higher level permissions can also grant this permission.

(Although similar functionality exists in the IRD Schema command language, this attribute-type is not associated with the IRD-SCHEMA-PERMISSIONS attribute-group-type. IRD Schema/IRD integrity constraints necessitate that the IRDS-user be authorized IRD-SCHEMA-ADMINISTRATOR-PERMISSION to move meta-entities between IRD Schema life-cycle-phases.)"

;

ADD META-ENTITY PANEL-PERMISSION  
META-ENTITY-TYPE = ATTRIBUTE-TYPE  
WITH META-ATTRIBUTES  
ORIGIN = X3.138-1988 (Ch.3)  
FORMAT = STRING  
MINIMUM-ATTRIBUTE-LENGTH = 1  
MAXIMUM-ATTRIBUTE-LENGTH = 1  
PURPOSE =

"This attribute-type is associated with the entity-type of IRDS-USER. For a given IRDS-user entity, a YES in the attribute of this type indicates that the user can use the panel interface. Attributes of this type are meaningful only if the implementation provides a command language interface."

;

ADD META-ENTITY READ-PERMISSION  
META-ENTITY-TYPE = ATTRIBUTE-TYPE  
WITH META-ATTRIBUTES  
ORIGIN = X3.138-1988 (Ch.3)  
FORMAT = STRING  
MINIMUM-ATTRIBUTE-LENGTH = 1  
MAXIMUM-ATTRIBUTE-LENGTH = 1

PURPOSE =

"This attribute-type is associated with two attribute-group-types: IRD-PERMISSIONS and IRD-SCHEMA-PERMISSIONS. In an IRD-PERMISSIONS attribute-group, a YES in the attribute of this type indicates that the user is allowed to select and display entities of the type identified in the corresponding ENTITY-TYPE-NAME attribute.

Similarly, in an IRD-SCHEMA-PERMISSIONS attribute-group, a YES in the attribute of this type indicates that a user can select and display meta-entities of the type identified in the corresponding META-ENTITY-TYPE-NAME attribute."

;

/\* The following additional attribute-types support the IRDS entity-level security facilities. \*/

ADD META-ENTITY ACCESS-KEYS-PERMISSION  
 META-ENTITY-TYPE = ATTRIBUTE-TYPE  
 WITH META-ATTRIBUTES

ORIGIN = X3.138-1988 (Ch.3)

FORMAT = STRING

MINIMUM-ATTRIBUTE-LENGTH = 1

MAXIMUM-ATTRIBUTE-LENGTH = 1

PURPOSE =

"This permission is meaningful only if it is specified in an IRD-view for which the IRD-PARTITION-NAME attribute equals SECURITY. This permission is ignored if it is specified for any IRD-view not associated with the SECURITY partition.

If the attribute of this type is meaningful and is equal to YES, the associated IRDS-user is authorized to use the add access-key and delete access-key commands. This permission restricts the user to add and delete access-keys to IRD-views which are related to the corresponding IRDS-user entity.

Unrestricted use of the add access-key and delete access-key commands is granted to the IRDS-user when the effective IRD-view authorizes MODIFY-PERMISSION for entities of type IRD-VIEW and READ-PERMISSION for entities of type ACCESS-CONTROLLER. This combination of permissions overrides the more restricted use defined by this attribute-type."

;

ADD META-ENTITY ADD-AND-SECURE-PERMISSION  
 META-ENTITY-TYPE = ATTRIBUTE-TYPE  
 WITH META-ATTRIBUTES

ORIGIN = X3.138-1988 (Ch.3)

FORMAT = STRING

MINIMUM-ATTRIBUTE-LENGTH = 1

MAXIMUM-ATTRIBUTE-LENGTH = 1

PURPOSE =

"This attribute-type is associated with the attribute-group-type IRD-PERMISSIONS. It is meaningful only for entity-types where IRDS entity-level security is in effect. In an IRD-PERMISSIONS attribute-group, a YES in the attribute of this type indicates that the user

can add entities of the type identified in the corresponding ENTITY-TYPE-NAME attribute and assign an access-controller to the newly created entity within a single command.

NOTE: A value of YES for this attribute also implies normal ADD-PERMISSION for entities of the corresponding type. The converse is not true."

;

ADD META-ENTITY READ-KEY

META-ENTITY-TYPE = ATTRIBUTE-TYPE  
WITH META-ATTRIBUTES  
ORIGIN = X3.138-1988 (Ch.3)  
FORMAT = STRING  
MINIMUM-ATTRIBUTE-LENGTH = 1  
MAXIMUM-ATTRIBUTE-LENGTH = 10  
PICTURE = NNNNNNNNNNN  
PURPOSE =

"This attribute-type is associated with the entity-type IRD-VIEW. It implements read keys for the IRDS entity-level security facility. Attributes of this type are not accessible via normal IRD maintenance commands."

;

ADD META-ENTITY READ-LOCK

META-ENTITY-TYPE = ATTRIBUTE-TYPE  
WITH META-ATTRIBUTES  
ORIGIN = X3.138-1988 (Ch.3)  
FORMAT = STRING  
MINIMUM-ATTRIBUTE-LENGTH = 1  
MAXIMUM-ATTRIBUTE-LENGTH = 10  
PICTURE = NNNNNNNNNNN  
PURPOSE =

"This attribute-type is associated with the entity-type ACCESS-CONTROLLER. It implements read locks for the IRDS entity security facility. Attributes of this type are not accessible via normal IRD maintenance commands."

;

ADD META-ENTITY SECURE-ENTITY-PERMISSION

META-ENTITY-TYPE = ATTRIBUTE-TYPE  
WITH META-ATTRIBUTES  
ORIGIN = X3.138-1988 (Ch.3)  
FORMAT = STRING  
MINIMUM-ATTRIBUTE-LENGTH = 1  
MAXIMUM-ATTRIBUTE-LENGTH = 1  
PURPOSE =

"This attribute-type is associated with the attribute-group-type, IRD-PERMISSIONS. A YES in the attribute of this type indicates that the user is authorized to add, modify, and delete the security of entities of a stated type in IRD partition corresponding to the IRD-view in which the attribute-group exists."

;



```

ADD META-ENTITY WRITE-KEY
  META-ENTITY-TYPE = ATTRIBUTE-TYPE
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    FORMAT = STRING
    MINIMUM-ATTRIBUTE-LENGTH = 1
    MAXIMUM-ATTRIBUTE-LENGTH = 10
    PICTURE = NNNNNNNNNNNN
    PURPOSE =
  "This attribute-type is associated with the entity-type IRD-VIEW. It
  implements write keys for the IRDS entity security facility.
  Attributes of this type are not accessible via normal IRD maintenance
  commands."
  ;

```

```

ADD META-ENTITY WRITE-LOCK
  META-ENTITY-TYPE = ATTRIBUTE-TYPE
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    FORMAT = STRING
    MINIMUM-ATTRIBUTE-LENGTH = 1
    MAXIMUM-ATTRIBUTE-LENGTH = 10
    PICTURE = NNNNNNNNNNNN
    PURPOSE =
  "This attribute-type is associated with the entity-type ACCESS-
  CONTROLLER. It implements write locks for the IRDS entity security
  facility. Attributes of this type are not accessible via normal IRD
  maintenance commands."
  ;

```

/\* Attributes of many of the above attribute-types have only two allowable values, YES and NO. The following commands define and implement this validation. \*/

```

ADD META-RELATIONSHIP ACCESS-KEYS-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP ACCESS-KEYS-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP ADD-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP ADD-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP
  ADD-AND-SECURE-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP ADD-AND-SECURE-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP
  COMMAND-LANGUAGE-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP
  COMMAND-LANGUAGE-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP DELETE-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP DELETE-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP
  IRD-ADMINISTRATOR-PERMISSION USES VALUE-VALIDATION;

```

```
ADD META-RELATIONSHIP
    IRD-ADMINISTRATOR-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP IRD-RENAME-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP IRD-RENAME-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP MODIFY-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP MODIFY-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP MODIFY-PHASE-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP MODIFY-PHASE-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP PANEL-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP PANEL-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP READ-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP READ-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP
    IRD-SCHEMA-ADMINISTRATOR-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP
    IRD-SCHEMA-ADMINISTRATOR-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP
    IRD-SCHEMA-RENAME-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP
    IRD-SCHEMA-RENAME-PERMISSION USES YES-OR-NO-VALUE;

ADD META-RELATIONSHIP SECURE-ENTITY-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP SECURE-ENTITY-PERMISSION USES YES-OR-NO-VALUE;
```

#### 9.4 New Attribute-Group-Types

/\* The following IRD Schema commands define the attribute-group-types which supports the IRDS security facility. \*/

```
ADD META-ENTITY IRD-SCHEMA-PERMISSIONS
    META-ENTITY-TYPE = ATTRIBUTE-GROUP-TYPE
    WITH META-ATTRIBUTES
        ORIGIN = X3.138-1988 (Ch.3)
        SIGNIFICANT-ATTRIBUTES = 1
        PURPOSE =
```

"This attribute-group-type is associated with the IRD-SCHEMA-VIEW entity-type. Attribute-groups of this type within a given IRD-schema-view define those permissions which are to be granted against meta-entities within the IRD Schema life-cycle-phase identified by the IRD-SCHEMA-PHASE-NAME attribute of the IRD-schema-view.

This attribute-group-type has five associated attribute-types:

```
META-ENTITY-TYPE-NAME (the key attribute-type)
READ-PERMISSION
ADD-PERMISSION
```

MODIFY-PERMISSION  
DELETE-PERMISSION

The precedence among the permissions is as follows:

DELETE > MODIFY > ADD > READ

The precedence is such that if a higher permission is granted, all lower permissions shall also be granted.

Each attribute-group within an IRD-schema-view has a unique META-ENTITY-TYPE-NAME attribute. It is possible for two sets of permissions to be defined for a given meta-entity-type. This occurs when one IRD-SCHEMA-PERMISSIONS attribute-group has a key of ALL, and another has a specific meta-entity-type. In this case, the corresponding permissions are OR'ed. Thus the IRD-schema-view defined by the command:

```
ADD ENTITY SVIEW1
  ENTITY-TYPE = IRD-SCHEMA-VIEW
  WITH ATTRIBUTES
    IRD-SCHEMA-PHASE-NAME = UNCONTROLLED
    IRD-SCHEMA-PERMISSIONS =
      (ALL, YES, YES, NO, NO),
      (RELATIONSHIP-CLASS-TYPE, YES, NO, NO, NO);
```

would still authorize ADD-PERMISSION for RELATIONSHIP-CLASS-TYPE meta-entities."

;

/\* The following meta-relationships define the components of IRD-SCHEMA-PERMISSIONS. The actual values for the GROUP-POSITION meta-attributes is not significant. However, the relative order defined by them is significant. \*/

```
ADD META-RELATIONSHIP
  IRD-SCHEMA-PERMISSIONS CONTAINS META-ENTITY-TYPE-NAME
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    GROUP-POSITION = 10;
```

```
ADD META-RELATIONSHIP
  IRD-SCHEMA-PERMISSIONS CONTAINS READ-PERMISSION
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    GROUP-POSITION = 20;
```

```
ADD META-RELATIONSHIP
  IRD-SCHEMA-PERMISSIONS CONTAINS ADD-PERMISSION
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    GROUP-POSITION = 30;
```

```
ADD META-RELATIONSHIP
  IRD-SCHEMA-PERMISSIONS CONTAINS MODIFY-PERMISSION
  WITH META-ATTRIBUTES
```

ORIGIN = X3.138-1988 (Ch.3)  
GROUP-POSITION = 40;

ADD META-RELATIONSHIP  
  IRD-SCHEMA-PERMISSIONS CONTAINS DELETE-PERMISSION  
  WITH META-ATTRIBUTES  
    ORIGIN = X3.138-1988 (Ch.3)  
    GROUP-POSITION = 50;

/\* End of the definition of IRD-SCHEMA-PERMISSIONS. \*/

ADD META-ENTITY IRD-PERMISSIONS  
  META-ENTITY-TYPE = ATTRIBUTE-GROUP-TYPE  
  WITH META-ATTRIBUTES  
    ORIGIN = X3.138-1988 (Ch.3)  
    SIGNIFICANT-ATTRIBUTES = 1  
    PURPOSE =

"This attribute group-type is associated with the entity-type of IRD-VIEW. Attribute-groups of this type within a given IRD-view define those permissions which are to be granted against entities within the partition identified by the IRD-PARTITION-NAME attribute of the IRD-view.

Each attribute-group within an IRD-view has a unique ENTITY-TYPE-NAME attribute. It is possible for two sets of permissions to be defined for a given meta-entity-type. This occurs when one IRD-PERMISSIONS attribute-group has a key of ALL, and another has a specific meta-entity-type. In this case, the corresponding permissions are OR'ed.

This attribute-group-type has eight associated attribute-types:

ENTITY-TYPE-NAME (the key attribute-type)  
READ-PERMISSION  
ADD-PERMISSION  
MODIFY-PERMISSION  
DELETE-PERMISSION  
MODIFY-PHASE-PERMISSION  
ADD-AND-SECURE-PERMISSION  
SECURE-ENTITY-PERMISSION

The precedence among the permissions is as follows:

DELETE > MODIFY > ADD > READ  
ADD-AND-SECURE > ADD > READ  
MODIFY-PHASE > READ  
SECURE-ENTITY > READ

The precedence is such that if a higher permission is granted, all lower permissions shall also be granted."

;

/\* The following meta-relationships identify the component attribute-types of IRD-PERMISSIONS. The specific values for GROUP-POSITION are not significant. However, the relative order of the component attribute-types is significant. \*/

```
ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS ENTITY-TYPE-NAME
  WITH META-ATTRIBUTES
    GROUP-POSITION = 10
    ORIGIN = X3.138-1988 (Ch.3)
  ;

ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS READ-PERMISSION
  WITH META-ATTRIBUTES
    GROUP-POSITION = 20
    ORIGIN = X3.138-1988 (Ch.3)
  ;

ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS ADD-PERMISSION
  WITH META-ATTRIBUTES
    GROUP-POSITION = 30
    ORIGIN = X3.138-1988 (Ch.3)
  ;

ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS MODIFY-PERMISSION
  WITH META-ATTRIBUTES
    GROUP-POSITION = 40
    ORIGIN = X3.138-1988 (Ch.3)
  ;

ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS DELETE-PERMISSION
  WITH META-ATTRIBUTES
    GROUP-POSITION = 50
    ORIGIN = X3.138-1988 (Ch.3)
  ;

ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS MODIFY-PHASE-PERMISSION
  WITH META-ATTRIBUTES
    GROUP-POSITION = 60
    ORIGIN = X3.138-1988 (Ch.3)
  ;

ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS ADD-AND-SECURE-PERMISSION
  WITH META-ATTRIBUTES
    GROUP-POSITION = 70
    ORIGIN = X3.138-1988 (Ch.3)
  ;
```

```
ADD META-RELATIONSHIP
  IRD-PERMISSIONS CONTAINS SECURE-ENTITY-PERMISSION
  WITH META-ATTRIBUTES
    GROUP-POSITION = 80
    ORIGIN = X3.138-1988 (Ch.3)
  ;
```

## 9.5 Changes to Entity-Types

/\* The following commands update the PURPOSE meta-attributes for existing entity-types in order to reflect their usage by this Module. Since these entity-types are substantially changed in scope, new versions are created. \*/

```
MODIFY META-ENTITY IRDS-USER
  NEW VERSION
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    PURPOSE (FIRST THROUGH LAST) =
  "Entities of this type are used to identify users to the IRDS. These
  entities also identify global permissions which are granted to a
  user. In particular, the following permissions are defined in
  entities of this type:
```

```
  IRD-ADMINISTRATOR-PERMISSION
  IRD-RENAME-PERMISSION
  IRD-SCHEMA-ADMINISTRATOR-PERMISSION
  IRD-SCHEMA-RENAME-PERMISSION
  COMMAND-LANGUAGE-PERMISSION
  PANEL-PERMISSION
```

The precedence among these permissions is as follows:

```
  IRD-ADMINISTRATOR > IRD-RENAME
  IRD-ADMINISTRATOR > COMMAND-LANGUAGE
  IRD-ADMINISTRATOR > PANEL
  IRD-SCHEMA-ADMINISTRATOR > IRD-SCHEMA-RENAME
  IRD-SCHEMA-ADMINISTRATOR > COMMAND-LANGUAGE
  IRD-SCHEMA-ADMINISTRATOR > PANEL
```

The precedence is such that if a higher permission is granted, all lower permissions shall also be granted."

```
  ;

MODIFY META-ENTITY IRD-VIEW
  NEW VERSION
  WITH META-ATTRIBUTES
    ORIGIN = X3.138-1988 (Ch.3)
    PURPOSE (FIRST THROUGH LAST) =
  "Entities of this type are used to define what actions can be taken
  within a given IRD-partition. Attribute-groups of type IRD-
  PERMISSIONS define permissions on an entity-type by entity-type
  basis. Additionally, two permissions are defined directly as
```

attributes of each entity of this type, ACCESS-KEYS-PERMISSION and IRD-ADMINISTRATOR-PERMISSION.

ACCESS-KEYS-PERMISSION is meaningful only if the specific IRD-view has an attribute of type IRD-PARTITION-NAME equal to SECURITY. This permission is independent of all permissions defined in the IRD-PERMISSIONS attribute-group.

The IRD-ADMINISTRATOR-PERMISSION attribute within an entity of this type has higher precedence than both ACCESS-KEYS-PERMISSION and all IRD-PERMISSIONS attribute groups."

;

MODIFY META-ENTITY IRD-SCHEMA-VIEW

NEW VERSION

WITH META-ATTRIBUTES

ORIGIN = X3.138-1988 (Ch.3)

PURPOSE (FIRST THROUGH LAST) =

"Entities of this type identify what actions are permitted within a given IRD Schema life-cycle-phase. These permissions are defined on a meta-entity-type by meta-entity-type basis within attribute-groups of type IRD-SCHEMA-PERMISSIONS. The attribute-type IRD-SCHEMA-ADMINISTRATOR-PERMISSION is also associated with this entity-type. This permission has higher precedence than IRD-SCHEMA-PERMISSIONS. For any given IRD-schema-view, IRD-SCHEMA-ADMINISTRATOR = YES is equivalent to IRD-SCHEMA-PERMISSIONS = ( META-ENTITY-TYPE-NAME = ALL, READ-PERMISSION = YES, ADD-PERMISSION = YES, MODIFY-PERMISSION = YES, DELETE-PERMISSION = YES )."

;

## 9.6 Entity-Type to Attribute-Type Meta-Relationships

/\* The following meta-relationships complete the IRDS-USER entity-type as required by this Module. \*/

ADD META-RELATIONSHIP

IRDS-USER CONTAINS COMMAND-LANGUAGE-PERMISSION

WITH META-ATTRIBUTES

SINGULAR = YES

ORIGIN = X3.138-1988 (Ch.3)

;

ADD META-RELATIONSHIP

IRDS-USER CONTAINS IRD-ADMINISTRATOR-PERMISSION

WITH META-ATTRIBUTES

SINGULAR = YES

ORIGIN = X3.138-1988 (Ch.3)

;

ADD META-RELATIONSHIP

IRDS-USER CONTAINS IRD-RENAME-PERMISSION

WITH META-ATTRIBUTES

SINGULAR = YES  
ORIGIN = X3.138-1988 (Ch.3)

;

ADD META-RELATIONSHIP  
IRDS-USER CONTAINS PANEL-PERMISSION  
WITH META-ATTRIBUTES  
SINGULAR = YES  
ORIGIN = X3.138-1988 (Ch.3)

;

ADD META-RELATIONSHIP  
IRDS-USER CONTAINS IRD-SCHEMA-ADMINISTRATOR-PERMISSION  
WITH META-ATTRIBUTES  
SINGULAR = YES  
ORIGIN = X3.138-1988 (Ch.3)

;

ADD META-RELATIONSHIP  
IRDS-USER CONTAINS IRD-SCHEMA-RENAME-PERMISSION  
WITH META-ATTRIBUTES  
SINGULAR = YES  
ORIGIN = X3.138-1988 (Ch.3)

;

/\* The following meta-relationships associate the new attribute-types to the IRD-VIEW entity-type as required by the IRDS Global Security Facilities. \*/

ADD META-RELATIONSHIP  
IRD-VIEW CONTAINS IRD-ADMINISTRATOR-PERMISSION  
WITH META-ATTRIBUTES  
SINGULAR = YES  
ORIGIN = X3.138-1988 (Ch.3)

;

ADD META-RELATIONSHIP  
IRD-VIEW CONTAINS EXCLUDE-RELATIONSHIPS  
WITH META-ATTRIBUTES  
SINGULAR = NO  
MAXIMUM-NUMBER-OF-OCCURRENCES = 256  
/\* MAX-OCC is Implementation Dependent \*/  
ORIGIN = X3.138-1988 (Ch.3)

;

/\* The following IRD Schema commands are required to implement locks and keys as required by the IRDS Entity-Level Security Facility. \*/

ADD META-RELATIONSHIP ACCESS-CONTROLLER CONTAINS READ-LOCK  
WITH META-ATTRIBUTES  
SINGULAR = YES  
ORIGIN = X3.138-1988 (Ch.3)

;



```

ADD META-RELATIONSHIP ACCESS-CONTROLLER CONTAINS WRITE-LOCK
    WITH META-ATTRIBUTES
        SINGULAR = YES
        ORIGIN = X3.138-1988 (Ch.3)
    ;

```

```

ADD META-RELATIONSHIP
    IRD-VIEW CONTAINS ACCESS-KEYS-PERMISSION
    WITH META-ATTRIBUTES
        SINGULAR = YES
        ORIGIN = X3.138-1988 (Ch.3)
    ;

```

```

ADD META-RELATIONSHIP IRD-VIEW CONTAINS READ-KEY
    WITH META-ATTRIBUTES
        SINGULAR = NO
        MAXIMUM-NUMBER-OF-OCCURRENCES = 32767
        /* MAX-OCC is Implementation Dependent */
        ORIGIN = X3.138-1988 (Ch.3)
    ;

```

```

ADD META-RELATIONSHIP IRD-VIEW CONTAINS WRITE-KEY
    WITH META-ATTRIBUTES
        SINGULAR = NO
        MAXIMUM-NUMBER-OF-OCCURRENCES = 32767
        /* MAX-OCC is Implementation Dependent */
        ORIGIN = X3.138-1988 (Ch.3)
    ;

```

/\* The following meta-relationships associate the new attribute-types with the IRD-SCHEMA-VIEW entity-type as required by this Module. \*/

```

ADD META-RELATIONSHIP
    IRD-SCHEMA-VIEW CONTAINS IRD-SCHEMA-ADMINISTRATOR-PERMISSION
    WITH META-ATTRIBUTES
        SINGULAR = YES
        ORIGIN = X3.138-1988 (Ch.3)
    ;

```

```

ADD META-RELATIONSHIP
    IRD-SCHEMA-VIEW CONTAINS EXCLUDE-META-RELATIONSHIPS
    WITH META-ATTRIBUTES
        SINGULAR = NO
        MAXIMUM-NUMBER-OF-OCCURRENCES = 256
        /* MAX-OCC is Implementation Dependent */
        ORIGIN = X3.138-1988 (Ch.3)
    ;

```

### 9.7 Entity-Type to Attribute-Group-Type Meta-Relationships

/\* The following IRD Schema commands specify the meta-relationships required to support the IRDS entity-level security facility. \*/

```
ADD META-RELATIONSHIP
  IRD-VIEW CONTAINS IRD-PERMISSIONS
  WITH META-ATTRIBUTES
    SINGULAR = NO
    MAXIMUM-NUMBER-OF-OCCURRENCES = 256
    /* MAX-OCC is Implementation Dependent */
    ORIGIN = X3.138-1988 (Ch.3)
;
```

```
ADD META-RELATIONSHIP
  IRD-SCHEMA-VIEW CONTAINS IRD-SCHEMA-PERMISSIONS
  WITH META-ATTRIBUTES
    SINGULAR = NO
    MAXIMUM-NUMBER-OF-OCCURRENCES = 256
    /* MAX-OCC is Implementation Dependent */
    ORIGIN = X3.138-1988 (Ch.3)
;
```

### 9.8 Modification to Meta-Entity of Type IRDS-Reserved Names

The meta-entity STANDARD-RESERVED-NAMES shall be modified by the following IRD Schema command:

```
MODIFY META-ENTITY STANDARD-RESERVED-NAMES
  META-ENTITY-TYPE = IRDS-RESERVED-NAMES
  WITH META-ATTRIBUTES
    RESERVED-META-ENTITY-NAME =
      ACCESS-CONTROLLER,
      ACCESS-KEYS-PERMISSION,
      ADD-PERMISSION,
      ADD-AND-SECURE-PERMISSION,
      COMMAND-LANGUAGE-PERMISSION,
      IRD-ADMINISTRATOR-PERMISSION,
      IRD-PERMISSIONS,
      IRD-RENAME-PERMISSION,
      DELETE-PERMISSION,
      ENTITY-TYPE-NAME,
      META-ENTITY-TYPE-NAME,
      MODIFY-PHASE-PERMISSION,
      MODIFY-PERMISSION,
      PANEL-PERMISSION,
      READ-PERMISSION,
      READ-KEY,
      READ-LOCK,
      IRD-SCHEMA-ADMINISTRATOR-PERMISSION,
      IRD-SCHEMA-PERMISSIONS,
      IRD-SCHEMA-RENAME-PERMISSION,
      SECURE-ENTITY-PERMISSION,
      SECURED-BY,
      WRITE-KEY,
      WRITE-LOCK
```

## 10 IRDS Security Panel Trees

**Function** To identify all panel trees which manage the security of individual entities.

### Format

/\* None. This breakdown is explanatory only. \*/

```

irds-security-panel-tree ::=
    add-security-panel-tree           10.1
    | modify-security-panel-tree      10.2
    | delete-security-panel-tree      10.3
    | add-access-key-panel-tree       10.4
    | delete-access-key-panel-tree    10.5

```

### Syntax Rules

- (1) The Syntax Rules which exist for user-supplied information in the Data Area shall include those which exist in the specifications for the equivalent Security command in the Command Language Interface. The Syntax Rules which shall not apply are those which relate to command-clause ordering within a particular command specification. Refer to the IRDS Security Commands, Section 6, for a description of these rules.

### General Rules

- (1) The General Rules which exist for these panel trees shall include those which exist for the equivalent command, with all of its associated command-clauses, in the Command Language Interface. Refer to Section 6 for a description of these rules.

### Security Rules

For all Security Rules in this and subordinate subsections:

Let  $U$  denote the effective IRDS-user.  
 Let  $V$  denote the effective IRD-view.

- (1) Each Security Rule specified for all IRDS Security Commands (Section 6) shall also apply to this Section. For each Panel Tree specified in a subordinate section, each of the Security Rules for the corresponding command shall apply.
- (2) Additionally,  $U$  shall be authorized PANEL-PERMISSION.

### Actions

- (1) The Actions which are performed by these panel trees shall be equivalent to those which are performed by the equivalent command in the Command Language Interface. Refer to Section 6 for a description of these actions.

### Error and Warning Conditions

- (1) Error and Warning Conditions which can appear in the Message Area of one or more of the panels in the panel trees described in the following subsections shall be the same as

those which exist for the equivalent irds-security-command in Section 6. Error and Warning Conditions which refer to improper ordering of command-clauses shall not apply to these panel trees.

### 10.1 Add Security Panel Tree

**Function** This panel tree is used to assign one or more access-controllers to one or more entities.

#### Data Area Contents

(1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:

- (a) To specify the entities to which access-controllers are to be assigned, one of the following shall be specified:
  - (i) The access-names of one or more existing entities which are to be assigned the specified access-controller.
  - (ii) The specification of entity selection criteria resulting in the selection of one or more entities. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

Additionally, the panel shall prompt for IRD-views to which the entity selection is to be limited. Entry of this information is optional, and the default is as specified in the entities-in-clause (Subsection 7.1 of this Module).

- (b) To specify the access-controllers which are to be assigned to the entities specified above, one of the following shall be specified:
  - (i) The access-names of one or more access-controllers.
  - (ii) The specification of entity selection criteria resulting in the selection of one or more access-controllers. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

#### IRD Schema Area Contents

(1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:

- (a) Rules for entity-naming.
- (b) The list of entity-types which can be assigned access-controllers.

### Action Area Contents

- (1) In addition to those global Action Area options specified in Subsection 8.3.4 of Module 1 of this standard, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Add security, in which case the specified access-controller shall be assigned to each specified or selected entity.
  - (b) Display validation rules, in which case the desired validation information will appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data will be erased from the Data Area.

### 10.2 Modify Security Panel Tree

**Function** This panel tree is used to deassign a specified access-controller from one or more entities and assign another in its place.

#### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) To specify the entities to which access-controllers are to be assigned, one of the following shall be specified:
    - (i) The access-names of one or more existing entities which are to be assigned the specified access-controller.
    - (ii) The specification of entity selection criteria resulting in the selection of one or more entities. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

Additionally, the panel shall prompt for IRD-views to which the entity selection is to be limited. Entry of this information is optional, and the default is as specified in the entities-in-clause (Subsection 7.1 of this Module).

- (b) The name of an existing access-controller to be deassigned from the specified or selected entities.
- (c) The name of a new or existing access-controller to be assigned to the specified or selected entities. If a new access-controller name is specified, the user shall be notified that a new access-controller is being created.

#### IRD Schema Area Contents

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:

- (a) Rules for entity-naming.
- (b) The list of entity-types which can be assigned access-controllers.

### Action Area Contents

- (1) In addition to those global Action Area options specified in Subsection 8.3.4 of Module 1 of this standard, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Modify security, in which case the specified access-controller to be deassigned is deassigned from each specified or selected entity, and the other access-controller entered is assigned to each specified or selected entity.
  - (b) Display validation rules, in which case the desired validation information will appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data will be erased from the Data Area.

### 10.3 Delete Security Panel Tree

**Function** This panel tree is used to remove either specific or all entity-level security constraints from one or more entities.

### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) To specify the entities from which access-controllers are to be deassigned, one of the following shall be specified:
    - (i) The access-names of one or more existing entities which are to be assigned the specified access-controller.
    - (ii) The specification of entity selection criteria resulting in the selection of one or more entities. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

Additionally, the panel shall prompt for IRD-views to which the entity selection is to be limited. Entry of this information is optional, and the default is as specified in the entities-in-clause (Subsection 7.1 of this Module).

- (b) To specify the access-controllers which are to be deassigned from the entities specified above, one of the following shall be specified:
  - (i) The access-names of one or more access-controllers. The user can optionally specify "ALL", in which case all applicable access-controllers will be disassociated from the specified or selected entities.

- (ii) The specification of entity selection criteria resulting in the selection of one or more access-controllers. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity-naming.
  - (b) The list of entity-types which can be assigned access-controllers.

### **Action Area Contents**

- (1) In addition to those global Action Area options specified in Subsection 8.3.4 of Module 1 of this standard, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Delete security, in which case for each specified or selected entity the specified access-controllers shall be deassigned. In the event the user specified "ALL" access-controllers, all access-controllers assigned to the specified or selected entities are deassigned.
  - (b) Display validation rules, in which case the desired validation information will appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data will be erased from the Data Area.

## **10.4 Add Access Key Panel Tree**

**Function** This panel tree is used to assign the read or write access-keys corresponding to one or more access-controllers to one or more IRD-views.

### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) A read or write option. If the read option is specified, a read access-keys will be assigned. If the write option is specified, write access-keys will be assigned.
  - (b) To specify the access-controllers for which corresponding access-keys are to be assigned, one of the following shall be specified:
    - (i) The access-names of one or more access-controllers.
    - (ii) The specification of entity selection criteria resulting in the selection of one or more access-controllers. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

- (c) To specify the IRD-views which are to be assigned the access-keys for the access-controllers specified in (b) above, one of the following shall be specified:
  - (i) The access-names of one or more IRD-views.
  - (ii) The specification of entity selection criteria resulting in the selection of one or more IRD-views. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

#### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the Rules for entity-naming to the user.

#### **Action Area Contents**

- (1) In addition to those global Action Area options specified in Subsection 8.3.4 of Module 1 of this standard, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Add access-key, in which case for each access-controller / IRD-view pair specified, the read or write access-key shall be assigned to the IRD-view.
  - (b) Display validation rules, in which case the desired validation information will appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data will be erased from the Data Area.

### **10.5 Delete Access Key Panel Tree**

**Function** This panel tree is used to deassign all or selected access-keys for one or more IRD-views.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) To specify the access-controllers for which corresponding access-keys are to be deassigned, one of the following shall be specified:
    - (i) The access-names of one or more access-controllers or ALL.
    - (ii) The specification of entity selection criteria resulting in the selection of one or more access-controllers. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.
  - (b) To specify the IRD-views which are to have the access-keys for the access-controllers specified in (a) above deassigned, one of the following shall be specified:



- (i) The access-names of one or more IRD-views.
- (ii) The specification of entity selection criteria resulting in the selection of one or more IRD-views. The entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33 of Module 1 of this standard.

### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the Rules for entity naming to the user.

### **Action Area Contents**

- (1) In addition to those global Action Area options specified in Subsection 8.3.4 of Module 1 of this standard, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Delete access-key, in which case for each access-controller/ IRD-view pair specified, the access-key shall be deassigned from the IRD-view.
  - (b) Display validation rules, in which case the desired validation information will appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data will be erased from the Data Area.

## **11 Cross Reference Indices**

Three cross-reference indices are provided:

- (1) A command index.
- (2) A command-clause index.
- (3) An error and warning condition index.

The command and command-clause indices provide an alphabetic cross reference of commands and command-clauses, respectively. The error and warning condition index is a usage index. It identifies where each error and warning condition is used in this Module. The scope of each of these indices is for this Module only.

### **11.1 Alphabetic Index of Commands**

<u>Command</u>	<u>Subsection</u>
add-access-key-command	6.4
add-security-command	6.1
delete-access-key-command	6.5
delete-security-command	6.3
modify-security-command	6.2

**11.2 Alphabetic Index of Command-Clauses**

<u>Command-Clause</u>	<u>Subsection Number</u>
assign-security-command-clause	7.8
controller-list-command-clause	7.5
IRD-view-in-command-clause	7.10
IRD-views-list-command-clause	7.6
entities-in-command-clause	7.1
exclude-security-command-clause	7.9
for-all-controllers-command-clause	7.7
for-controllers-in-command-clause	7.11
from-controller-command-clause	7.2
related-controllers-command-clause	7.4
to-controller-command-clause	7.3

**11.3 Error and Warning Condition Usage Index**

<u>Cond</u>	<u>Text / Subsection</u>
E01009	Entity does not exist or is not visible. 6.1      6.2      6.3      7.10      7.11
E01067	Meta-entity does not exist or is not visible. 8.1.1.2    8.1.1.10
W01037	Creation of corresponding meta-relationship would violate IRD Schema life-cycle-phase integrity. 8.1.1.2    8.1.1.10
E03001	IRDS-user has no corresponding IRD-schema-view. 8.1
E03002	Effective IRD-schema-view does not authorize user to use command. 8.1.1.1    8.1.1.2    8.1.1.3    8.1.1.4    8.1.1.5    8.1.1.6 8.1.1.10
E03003	User not authorized to delete a meta-relationship in which specified meta-entity participates. 8.1.1.3
E03004	User not authorized to perform specified action(s). 6.1      6.2      6.3      8.1.1.4    8.1.1.5    8.1.1.6 8.1.1.7    8.1.1.8    8.1.1.9    8.1.1.11    8.1.1.12    8.1.1.13 8.2.1.2    8.2.1.3    8.2.1.4    8.2.1.5    8.2.1.6    8.2.1.7 8.2.1.8    8.2.1.9    8.2.1.10    8.3.6
E03005	IRDS-user has no corresponding IRD-view. 8.2
E03006	IRD-view does not identify a valid IRD partition. 8.2

- E03007 Cannot secure a security entity.  
6.1 6.2 6.3 8.2.1.1 8.2.1.10
- E03008 Entity-level security cannot be assigned to specified entity.  
6.1 6.2 6.3 8.2.1.1
- E03009 System-maintained attributes cannot be specified in maintenance commands.  
8.2.1.1 8.2.1.2
- E03011 User not authorized to perform specified action(s) by the effective IRD-view.  
6.1 6.2 6.3 6.4 6.5 8.2.1.1  
8.2.1.2 8.2.1.3 8.2.1.9 8.2.1.10 8.3.4
- E03012 Relationships which implement entity-level security cannot be maintained by normal maintenance commands.  
8.2.1.4 8.2.1.5 8.2.1.6
- E03013 An access-controller cannot be copied.  
8.2.1.10
- E03014 IRD Schema descriptor not visible to user.  
8.2.2
- E03015 User does not have adequate authorization to source IRD Schema.  
8.3.1 8.3.2
- E03016 User does not have adequate authorization to current or specified IRD Schema.  
8.3.3
- E03017 Effective IRD-view must be associated with SECURITY partition.  
6.0
- E03018 From-controller not assigned to entity.  
6.2
- E03019 To-controller already assigned to entity.  
6.2
- E03020 Entity is not an access-controller.  
6.4 6.5 7.2 7.3 7.4 7.5  
7.8
- E03021 Entity is not an IRD-view.  
6.4 6.5 7.6
- E03022 Command not authorized against specified or selected IRD-view.  
6.4 6.5
- E03023 Access-controller-name does not exist in the IRD.  
7.2 7.3 7.4 7.5 7.8
- E03024 IRD-view-name does not exist in the IRD.  
7.6

- E03025    User cannot assign/deassign specified access-controller.  
6.1            6.2            6.3
  
- E03026    Invalid IRD-view for command-clause. Specified IRD-view associated with SECURITY IRD-partition.  
7.1
  
- E03027    Invalid IRDS-user for command-clause. Specified IRD-view associated must be associated with SECURITY IRD-partition.  
7.10            7.11
  
- W03001    User not authorized to create corresponding meta-relationship.  
8.1.1.1.2    8.1.1.1.10
  
- W03002    Entity-level security not defined for entities of this type.  
8.2.1.2
  
- W03003    Impacted entity not visible.  
8.2.2.1
  
- W03004    Entity already assigned specified access-controller.  
6.1
  
- W03005    Specified or selected entity is still secured by access-controller not in the user's security maintenance set.  
6.3
  
- W03006    Specified or selected entity is not secured.  
6.3
  
- W03007    Specified access-controller is not assigned to entity.  
6.3

# Appendix

(This Appendix is not part of American National Standard X3.138-1988, but is included for information only.)

## Collected Syntax

access-controller-name ::= assigned-access-name

access-key-word ::= ACCESS-KEY

access-keys-word ::=  
ACCESS-KEYS  
| access-key-word

access-name-list ::=  
access-name  
[ , access-name-list ]

add-access-key-command ::=  
add-word  
read-or-write-option  
access-keys-word  
controller-selection-option  
to-word  
IRD-views-option  
;

add-entity-command ::=  
add-word  
entity-word  
new-entity-name  
entity-type-clause  
[ entity-descriptive-name-declaration-clause ]  
[ quality-indicator-designation-clause ]  
[ assign-security-clause ]  
[ new-entity-attributes-clause ]  
;

add-security-command ::=  
add-word  
security-word  
[ to-word ]  
secured-entities-option  
controller-list-clause  
;

assign-security-clause ::=  
assign-word  
security-word  
[ controller-word  
=  
]  
access-controller-name

assign-word ::= ASSIGN

controller-identification ::=  
all-word  
| controller-name-list

controller-list-clause ::=  
controller-word  
=  
controller-name-list

controller-name-list ::=  
access-controller-name  
[ , controller-name-list ]

controller-selection-option ::=  
for-all-controllers-clause  
| controller-list-clause  
| for-controllers-in-clause

controller-word ::=  
ACCESS-CONTROLLER  
| CONTROLLER

controllers-word ::=  
controller-word  
| ACCESS-CONTROLLERS  
| CONTROLLERS

APPENDIX

```

copy-entity-command ::=
    copy-word
    entity-word
    existing-entity-access-name
    [ with-relationships-clause ]
    to-word
    new-entity-access-name
    [ entity-descriptive-name-declaration-clause ]
    [ quality-indicator-designation-clause ]
    [ exclude-security-clause ]
    [ assign-security-clause ]
    ;

delete-access-key-command ::=
    delete-word
    access-keys-word
    controller-selection-option
    to-word
    IRD-views-option
    ;

delete-security-command ::=
    delete-word
    security-word
    [ of-word ]
    secured-entities-option
    related-controllers-clause
    ;

entities-in-clause ::=
    entities-word
    in-word
    entity-selection-set-spec

entity-selection-set-spec ::=
    [ using-IRD-views-clause ]
    entity-selection-criteria-clause
    | entity-selection-criteria-clause
    [ using-IRD-views-clause ]
    | using-list-clause

exclude-security-clause ::=
    exclude-word
    security-word

existing-entity-name ::=
    access-name

for-all-controllers-clause ::=
    for-word
    all-word
    controllers-word

for-controllers-in-clause ::=
    for-word
    controllers-word
    in-word
    entity-selection-set-spec

from-controller-clause ::=
    from-word
    [ controller-word ]
    access-controller-name

IRD-view-name ::=
    assigned-access-name

IRD-view-name-list ::=
    IRD-view-name
    [ , IRD-view-name-list ]

IRD-views-in-clause ::=
    IRD-views-word
    in-word
    entity-selection-set-spec

IRD-views-list-clause ::=
    IRD-view-word
    =
    IRD-view-name-list

IRD-views-option ::=
    IRD-views-list-clause
    | IRD-views-in-clause

irds-security-command ::=
    add-security-command
    | modify-security-command
    | delete-security-command
    | add-access-key
    | delete-access-key

modify-entity-command ::=
    modify-word
    entity-word
    existing-entity-name
    [ entity-type-clause ]
    [ new-version-clause
    [ exclude-security-clause ]
    ]
    [ entity-descriptive-name-declaration-clause ]
    [ quality-indicator-designation-clause ]
    [ modified-entity-attributes-clause ]
    ;

```

```

modify-entity-life-cycle-phase-command ::=
    modify-word
    entity-word
    life-cycle-phase-word
    [ for-word ]
    entity-name-list
    from-word
    current-life-cycle-phase
    to-word
    new-life-cycle-phase
    [ new-version-clause
      [ exclude-security-clause ]
    ]
    [ assign-security-clause ]
    ;

modify-security-command ::=
    modify-word
    security-word
    [ of-word ]
    secured-entities-option
    from-controller-clause
    to-controller-clause
    ;

new-entity-access-name ::=
    access-name
    | null-mark
    | new-version-clause

new-entity-name ::=
    access-name
    | null-mark

read-or-write-option ::=
    read-word
    | write-word

read-word ::= READ

related-controllers-clause ::=
    controllers-word
    =
    controller-identification

secured-entities-option ::=
    access-name-list
    | entities-in-clause

security-word ::= SECURITY

to-controller-clause ::=
    to-word
    [ controller-word ]
    access-controller-name

write-word ::= WRITE

```





**Chapter 4****Module 4 - Extensible Life Cycle Phase Facility****CONTENTS**

<b>SECTION</b>	<b>PAGE</b>
1 Scope, Purpose, and Application . . . . .	4-1
1.1 Scope . . . . .	4-1
1.2 Purpose . . . . .	4-1
1.3 Conformance . . . . .	4-2
1.4 Organization . . . . .	4-2
1.5 Notation . . . . .	4-3
2 References . . . . .	4-5
3 Definitions . . . . .	4-5
4 Module Overview . . . . .	4-6
4.1 Summary of New Functionality . . . . .	4-6
4.2 Basic Concepts . . . . .	4-7
4.2.1 The Life-Cycle-Phase Hierarchy . . . . .	4-7
4.2.1.1 Implementation in the IRD Schema . . . . .	4-8
4.2.1.2 Integrity Constraints on the Life-Cycle-Phase-Hierarchy . . . . .	4-8
4.2.1.3 Implications on IRD Maintenance . . . . .	4-9
4.2.2 IRD-Schema-Structures . . . . .	4-9
4.2.2.1 Implementation in the IRD Schema . . . . .	4-9
4.2.2.2 IRD-Schema-Structure Subsets . . . . .	4-10
4.2.2.3 Integrity Constraints for IRD Schema Structures . . . . .	4-10
4.2.3 The Phase Sensitivity Structure . . . . .	4-11
4.2.4 IRD Life-Cycle-Phase Integrity . . . . .	4-12
4.2.5 The IRD-Schema-Structure IRD-STRUCTURE . . . . .	4-13
5 Basic Language Elements . . . . .	4-13
5.1 Function . . . . .	4-13
5.2 Format . . . . .	4-13
5.3 Syntax Rules . . . . .	4-14
5.4 General Rules . . . . .	4-14
6 IRD Schema Structure Manipulation Commands . . . . .	4-14
6.1 Add IRD-Schema-Structure Command . . . . .	4-15
6.2 Modify IRD-Schema-Structure Command . . . . .	4-18
6.3 Delete IRD-Schema-Structure Command . . . . .	4-23
6.4 Copy IRD-Schema-Structure Command . . . . .	4-25
6.5 Install IRD-Structure Command . . . . .	4-28
7 Command-Clauses . . . . .	4-30
7.1 Append Command-Clause . . . . .	4-30
7.2 Include Command-Clause . . . . .	4-31
7.3 Disconnect Command-Clause . . . . .	4-31

SECTION	PAGE
7.4 Exclude Command-Clause . . . . .	4-32
7.5 Members Command-Clause . . . . .	4-33
7.6 Subsets Command-Clause . . . . .	4-33
8 IRD Schema Structure Manipulation Panel Trees . . . . .	4-34
9 Modifications to the IRD Schema Definition . . . . .	4-35
9.1 New Meta-Entity-Types . . . . .	4-35
9.2 New Meta-Relationship-Types . . . . .	4-35
10 Modifications to Module 1 . . . . .	4-37
10.1 Modifications to IRD Schema Commands . . . . .	4-38
10.1.1 Modifications to Add Meta-Entity Command . . . . .	4-39
10.1.2 Modifications to Modify Meta-Entity Command . . . . .	4-39
10.1.3 Modifications to Delete Meta-Entity Command . . . . .	4-39
10.1.4 Modifications to Add Meta-Relationship Command . . . . .	4-39
10.1.5 Modifications to Modify Meta-Relationship Command . . . . .	4-39
10.1.6 Modifications to Delete Meta-Relationship Command . . . . .	4-40
10.1.7 Modifications to Modify Meta-Entity Life-Cycle-Phase Command . . . . .	4-40
10.1.8 Modifications to Copy Meta-Entity Command . . . . .	4-40
10.1.9 Modifications to Activate IRD Command . . . . .	4-41
10.2 Modifications to the IRD Commands . . . . .	4-41
10.2.1 Modifications to Add Entity Command . . . . .	4-41
10.2.2 Modifications to Modify Entity Command . . . . .	4-42
10.2.3 Modifications to Delete Entity Command . . . . .	4-43
10.2.4 Modifications to Add Relationship Command . . . . .	4-44
10.2.5 Modifications to Modify Relationship Command . . . . .	4-44
10.2.6 Modifications to Delete Relationship Command . . . . .	4-45
10.2.7 Modifications to Modify Entity Life-Cycle-Phase Command . . . . .	4-45
10.2.8 Modifications to Copy Entity Command . . . . .	4-46
10.2.9 Output Impact-of-Change Command . . . . .	4-47
10.3 Modifications to IRD-IRD Interface Commands . . . . .	4-47
10.3.1 Modifications to Create IRD Command . . . . .	4-47
10.3.2 Modifications to Check IRD Schema Compatibility Command . . . . .	4-48
10.4 Modifications in Command-Clauses for Module 1 Commands . . . . .	4-48
10.4.1 Modifications to With Relationships Command-Clause . . . . .	4-48
10.4.2 Modifications to Life-Cycle-Phase Restriction . . . . .	4-48
10.5 Modifications to the Minimal IRD Schema . . . . .	4-49
10.6 Conversion Requirements . . . . .	4-50
11 Modifications to Module 2 . . . . .	4-51
12 Modifications to Module 3 . . . . .	4-51
13 Cross-reference Indices . . . . .	4-51
13.1 Alphabetic Index of Commands . . . . .	4-51
13.2 Alphabetic Index of Command-Clauses . . . . .	4-51
13.3 Error and Warning Condition Usage Index . . . . .	4-52

<b>SECTION</b>	<b>PAGE</b>
<b>Appendixes</b>	
Appendix A - Collected Syntax . . . . .	4-55
Appendix B - Examples of Schema-Structure Manipulation . . . . .	4-56
<b>Figures</b>	
Figure 1 - Schema Structure for Example 1 . . . . .	4-62
Figure 2 - Schema Structures for Example 2 . . . . .	4-63
Figure 3 - Schema Structures for Example 3 . . . . .	4-64
Figure 4 - Schema Structures for Example 4 . . . . .	4-65
Figure 5 - Schema Structures for Example 5 . . . . .	4-66
Figure 6 - Schema Structures for Example 6 . . . . .	4-67
Figure 7 - Schema Structures for Example 7 . . . . .	4-68
Figure 8 - Schema Structures for Example 8 . . . . .	4-69
Figure 10 - Schema Structures for Example 10 . . . . .	4-71



## Chapter 4

### Module 4 - Extensible Life Cycle Phase Facility

#### 1 Scope, Purpose, and Application

##### 1.1 Scope

Module 4 of the standard IRDS specifies facilities which are appropriate specifically to control of an installation's information resources as they progress through a life cycle. This Module assumes the existence of an implementation of Module 1 of the standard IRDS.

##### 1.2 Purpose

This specification of the extensible life cycle phase facility provides the basis for life cycle management of the contents of the Information Resource Dictionary (IRD). This facility is appropriate for all entities and relationships in the IRD, except for those defined by the Minimal IRD Schema. Thus, an installation can establish life cycle management, as described herein, over the entities and relationships which were either defined by an implementor-provided IRD Schema or by extensions to the IRD Schema made by the installation.

The facility specified by this Module is similar to that provided for life cycle management of the IRD Schema, as specified in Module 1 of the standard IRDS. Similarities are:

- (1) The three basic classes of life-cycle-phases are the same; i.e., UNCONTROLLED, CONTROLLED, and ARCHIVED.
- (2) There can be only one CONTROLLED and one ARCHIVED life-cycle-phase.
- (3) A command exists for moving entities from one life-cycle-phase to another, with attendant versioning. The rules for assignment of Version Identifiers, are also the same.

There are some differences, however. These are due primarily to the fact that there is currently no specification for an IRD Schema definition level in this standard. These dissimilarities are:

- (1) There is an embedded life-cycle-phase control structure defined for the meta-entity-types which exist in the IRD Schema. This structure is based on the semantics of the entity-relationship model used in this specification. For the IRD, on the other hand, the meaning of the structure is installation-dependent; it is associated with the meta-entities which the installation uses.

There is, therefore, a need for the installation to be able to identify the IRD life-cycle-phase control structure and to manage it in a manner acceptable to installation requirements. This Module provides the means for establishing a tree-structure

hierarchy for the life-cycle-phase meta-entities and for using this hierarchy in conjunction with the rules of modifying the life-cycle-phase of entities.

- (2) There are no facilities provided in Module 1 for specifying structures involving IRD Schema descriptors.
- (3) There are no facilities specified in Module 1 for automatic management of the IRD life-cycle-phase integrity constraints.
- (4) There can be multiple life-cycle-phases of class UNCONTROLLED for the IRD, but only one UNCONTROLLED IRD Schema life-cycle-phase.

An installation would use this Module as follows:

- (1) Using the IRD Schema maintenance facility, new entity-types and relationship-types (along with appropriate attribute-types and attribute-group-types) would be added to the IRD Schema.
- (2) Appropriate meta-relationships would be established among these meta-entities, and if required, existing ones. These meta-relationships define an implicit structure with regard to IRD content.
- (3) A tree-structured life-cycle-phase hierarchy would be specified.
- (4) A special IRD-schema-structure can be defined. This IRD-schema-structure identifies the relationship-types whose instances are subject to life cycle phase integrity constraints.
- (5) This IRD-schema-structure is made effective in the IRD Schema.

Based on the above framework, entities can be moved from UNCONTROLLED to CONTROLLED, CONTROLLED to ARCHIVED, and CONTROLLED to UNCONTROLLED using the modify entity-life-cycle-phase command. The IRDS will assure that movements are only accomplished based on the life-cycle-phase integrity constraints.

### **1.3 Conformance**

The Definition of a Conformant Implementation of Module 4 of a standard IRDS is given in the Definition of a Conformant Implementation of the IRDS (see Section 4 of Requirements for a Conformant Implementation).

### **1.4 Organization**

Module 4 of the standard IRDS consists of the following:

- (1) A Command Language.
- (2) Modifications to the Syntax and Rules of Module 1.
- (3) Modifications to the IRD Schema defined in Module 1.

The Command Language defined in this Module is an upward compatible extension to the Command Language defined in Module 1. Wherever appropriate, the syntax components defined in Module 1 are reused.

The modifications to the syntax and rules of Module 1 are in the form of:

- (1) Additional rules for specific commands.
- (2) Additional actions to be performed by specific commands.

The modifications to the Definition of the IRD Schema in Module 1 consist of:

- (1) A new meta-entity-type.
- (2) A set of new meta-relationship-types.

Definitions of terms for this Module are given in Section 3.

Section 4 provides the technical basis for this Module. It provides the following:

- (1) A summary of the new functionality provided by the facilities defined in this Module.
- (2) A discussion of the basic concepts and terminology used throughout this Module.

Section 5 defines the basic elements of the Command Language syntax. This section consists of new elements not defined in Module 1.

The new Commands are defined in Section 6. The format is the same as found in Module 1 for command descriptions.

The Command-Clauses which are unique to this Module are defined in Section 7.

A description of the panel trees required to implement the functionality of this Module in the Panel Interface are provided in Section 8.

The Modifications to the IRD Schema Definition are defined in Section 9.

The Impact on the Core Commands and Panel Trees, including required modifications, is identified in Section 10.

Modifications to Module 2 and Module 3 of this standard are provided in Sections 11 and 12, respectively.

Section 13 provides cross-reference indices for all commands, command-clauses, and error and warning conditions defined in this Module.

To aid the reader in reviewing this Chapter, Table 1 provides a suggested review sequence.

## 1.5 Notation

This Module uses the same notational conventions as identified in Module 1 of the standard IRDS, except as noted below.

The Commands and Command-Clauses defined in this Module have cross-references to syntax defined both in this Module and in other Modules. Whenever a cross reference number is given as a sequence of numbers alone, the reference is to a section or subsection within this Module. References to other Modules begin with  $M_n$ , where  $n$  denotes the number of the other Module of the standard IRDS.

**Table 1**  
**Suggested Review Sequence**

ORDER	SECTION	TITLE	ACTION
1	1	SCOPE, PURPOSE AND APPLICATION	READ
2	4	MODULE OVERVIEW	READ
3	2, 3, 5	REFERENCES, DEFINITIONS, BASIC LANGUAGE ELEMENTS	SKIM
4	6	IRD SCHEMA STRUCTURE MANIPULATION COMMANDS	READ
5	9	MODIFICATIONS TO THE STRUCTURE OF THE IRD SCHEMA	SKIM
6	10	MODIFICATIONS TO MODULE 1	READ
7	8	IRD SCHEMA STRUCTURE MANIPULATION PANEL TREES	READ
8	11	MODIFICATIONS TO MODULE 2	READ
9	12	MODIFICATIONS TO MODULE 3	READ
10	7	COMMAND-CLAUSES	AS REQUIRED
11	13	CROSS REFERENCE INDICES	AS REQUIRED

Thus:

"6.2" denotes a reference to a subsection in this Module;

whereas

"M1.4.3" is a reference to Subsection 4.3 of Module 1.

Error and warning conditions have the following format:

(1) One alphabetic character which defines the message level:

E identifies an error.  
W identifies a warning.



- (2) Two digits identify the number of the Module of the IRDS standard where the message first occurs.
- (3) Three digits identify the message number within the Module of the standard IRDS.

## 2 References

None.

## 3 Definitions

This set of definitions apply only to this Module.

**direct subset (of an IRD-schema-structure).** An IRD-schema-structure *S'* is a direct subset of an IRD-schema-structure *S* if and only if the meta-relationship *S HAS-SUBSET S'* exists.

**direct superset (of an IRD-schema-structure).** An IRD-schema-structure *S* is a direct superset of an IRD-schema-structure *S'* if and only if the meta-relationship *S HAS-SUBSET S'* exists.

**IRD life-cycle-phase.** An IRD-partition which has a meta-attribute of type LIFE-CYCLE-PHASE-CLASS equal to UNCONTROLLED, CONTROLLED or ARCHIVED.

**IRD life-cycle-phase integrity.** The rules which define what relationships can exist between entities in different IRD life-cycle-phases. See Subsection 4.2.4 for the detailed formal definition.

**IRD Structure.** The IRD Structure is the set of all entity-types and relationship-types whose instances can exist in the IRD. There exists one and only IRD-schema-structure in the CONTROLLED IRD Schema life-cycle-phase with an assigned-access-name IRD-STRUCTURE which defines this set for the IRDS.

**IRD-partition.** A logical subset of IRD entities. Each IRD entity exists in one and only one IRD-partition. For each IRD-partition there exists a meta-entity in the IRD Schema of type IRD-PARTITION.

**IRD-schema-structure.** A meta-entity of type IRD-SCHEMA-STRUCTURE. An IRD-schema-structure defines a set of entity-types and relationship-types such that if a relationship-type is a member of the set, each entity-type which participates in the relationship-type is also a member of the set.

**life-cycle.** A conceptual framework which is used to trace the evolution of objects of certain classes over time. A life-cycle is divided into phases. An object is identified as existing in a particular life-cycle-phase. An object shall exist in only one life-cycle-phase at a given instance in time.

**life-cycle-phase hierarchy (LCPH).** A tree structure composed of IRD life-cycle-phases which is defined in the IRD Schema by meta-entities of type IRD-PARTITION and meta-relationships of type IRD-PARTITION-GREATER-THAN-IRD-PARTITION.

**member (of an IRD-schema-structure).** A meta-entity *M* of type ENTITY-TYPE or RELATIONSHIP-TYPE is a member of an IRD-schema-structure *S* if and only if the meta-relationship *S* SET-OF *M* exists.

**Phase Sensitivity Structure.** The set of entity-types and relationship-types which are subject to IRD life-cycle-phase integrity constraints. Each entity-type which can exist in an IRD life-cycle-phase is a member of this IRD-schema-structure. This set is defined in the IRD Schema by an IRD-schema-structure in the CONTROLLED IRD Schema life-cycle-phase with an assigned-access-name of PHASE-SENSITIVITY-STRUCTURE. There exists one and only one such meta-entity. It can be empty, and it is a subset of the IRD Structure.

**phase-related relationship.** A relationship whose corresponding type is phase-related.

**phase-related relationship-type.** A relationship-type which is a member of the Phase Sensitivity Structure.

**subset (of an IRD-schema-structure).** An IRD-schema-structure *S'* is a subset of *S* if it is either a direct subset of *S* or if there is a sequence of IRD-schema-structures *S*<sub>1</sub> ... *S*<sub>*n*</sub> such that *S'* is a direct subset of *S*<sub>1</sub>, each *S*<sub>*i*</sub> is a direct subset of *S*<sub>*i*+1</sub>, and *S*<sub>*n*</sub> is a direct subset of *S*.

**superset (of an IRD-schema-structure).** An IRD-schema-structure *S* is a superset of *S'* if it is either a direct superset of *S'* or if there is a sequence of IRD-schema-structures *S*<sub>1</sub> ... *S*<sub>*n*</sub> such that *S* is a direct superset of *S*<sub>*n*</sub>, each *S*<sub>*i*</sub> is a direct superset of *S*<sub>*i*-1</sub>, and *S*<sub>1</sub> is a direct superset of *S'*.

## 4 Module Overview

This Module of the standard IRDS specifies facilities which allow an installation to enforce life-cycle-management controls over the IRD. Module 1 specifies a similar control facility which applies only to the IRD Schema.

### 4.1 Summary of New Functionality

This Module of the standard provides:

- (1) An extension to the IRD Schema Definition, which allows IRD-schema-structures to be specified; IRD Schema structures are sets of entity-types and relationship-types, and these sets, have special semantics. New commands have been added to facilitate the maintenance of IRD-schema-structures.

IRD-schema-structures provide:

- (a) A mechanism for defining life-cycle-phase integrity in the IRD for the entity-types and relationship-types used at a given installation. This is accomplished by specifying a special IRD Schema structure, called the Phase Sensitivity Structure, which contains all entity-types and relationship-types for which life-cycle-phase constraints apply. Module 1 IRD commands and panels are modified to provide life-cycle management controls similar to those found in the Module 1 IRD Schema commands.

- (b) A mechanism which simplifies the manner in which new IRD Schema descriptors are placed in the CONTROLLED IRD Schema life-cycle phase. A special IRD-schema-structure, called the IRD Structure, is used to specify all meta-entities which are to be in the CONTROLLED IRD Schema life-cycle-phase. A new command, the Install IRD Structure Command, simplifies the movement to and from the CONTROLLED IRD Schema life-cycle-phase.
  - (c) An aid to documenting characteristics of the IRD Schema.
- (2) A means for establishing a hierarchy of IRD life-cycle-phases. This hierarchy can be used to document the methodology employed at an installation.

## 4.2 Basic Concepts

The five key concepts found in this Module of the standard IRDS are:

- (1) The Life-Cycle-Phase Hierarchy
- (2) IRD Schema Structures
- (3) The IRD Schema Structure named Phase Sensitivity Structure
- (4) IRD Life-Cycle-Phase Integrity
- (5) The IRD Schema Structure named IRD Structure

The following Subsections discuss each one of these concepts in detail.

### 4.2.1 The Life-Cycle-Phase Hierarchy

In Module 1 of the standard IRDS, there is a life-cycle management facility which operates on meta-entities. This facility depends on three life-cycle-phases in the IRD Schema, which are named UNCONTROLLED, CONTROLLED, and ARCHIVED. These life-cycle-phases are ordered, and the ordering UNCONTROLLED < CONTROLLED < ARCHIVED expresses the order of movement of meta-entities through a life-cycle. The basic flow of meta-entities through this life cycle is:

- (1) All meta-entities which shall be subject to life-cycle management shall be initially created in the UNCONTROLLED IRD Schema life-cycle-phase.
- (2) At some point they can be "moved" to the CONTROLLED IRD Schema life-cycle-phase; when they move to this phase they define the current IRD structure and validation parameters.
- (3) Later, they can be made obsolete by new meta-entities. They would then be moved to the ARCHIVED IRD Schema life-cycle-phase.

Similar control is desired for the IRD; this Module provides a similar, but more generalized, set of capabilities for the IRD.

In the IRD, an IRD life-cycle-phase is defined using a meta-entity of type IRD-PARTITION with a LIFE-CYCLE-PHASE-CLASS meta-attribute, which has three possible values: UNCONTROLLED, CONTROLLED, and ARCHIVED. There is only one meta-entity of type IRD-PARTITION with a LIFE-CYCLE-PHASE-CLASS meta-attribute equal to:

- (1) ARCHIVED: This meta-entity is called the archived IRD life-cycle-phase.
- (2) CONTROLLED: This meta-entity is called the controlled IRD life-cycle-phase.

All other IRD-PARTITION meta-entities which define an IRD life-cycle-phase shall have a LIFE-CYCLE-PHASE-CLASS meta-attribute equal to UNCONTROLLED. These are called uncontrolled IRD life-cycle-phases.

The Life-Cycle-Phase Hierarchy is a tree-structure of IRD life-cycle-phases such that:

- (1) The archived IRD life-cycle-phase is the root of the tree.
- (2) The controlled IRD life-cycle-phase is directly subordinate to the archived IRD life-cycle-phase.
- (3) Each uncontrolled IRD life-cycle-phase is directly subordinate to only one other IRD life-cycle-phase. All uncontrolled IRD life-cycle-phases shall be (directly or indirectly) subordinate to the controlled IRD life-cycle-phase.

#### 4.2.1.1 Implementation in the IRD Schema

The Life-Cycle-Phase Hierarchy shall be defined in the IRD Schema using meta-relationships of type IRD-PARTITION-GREATER-THAN-IRD-PARTITION; this meta-relationship-type belongs to the meta-relationship-class-type GREATER-THAN.

In order to express the hierarchy, a given IRD life-cycle-phase  $P$  is said to be superior to another IRD life-cycle-phase  $P'$  if and only if one of the following is satisfied:

- (1) The meta-relationship  $P$  GREATER-THAN  $P'$  exists.
- (2) There exists a set of IRD life-cycle-phases  $P_1 \dots P_n$  such that the meta-relationships

$$\begin{array}{l} P \text{ GREATER-THAN } P_1 \\ \dots \\ P_i \text{ GREATER-THAN } P_{i+1} \\ \dots \\ P_n \text{ GREATER-THAN } P' \end{array}$$

exist.

In (1) above,  $P$  is said to be directly superior to  $P'$ . Conversely, if  $P$  is (directly) superior to  $P'$ ,  $P'$  is said to be (directly) subordinate to  $P$ .

#### 4.2.1.2 Integrity Constraints on the Life-Cycle-Phase-Hierarchy

A meta-entity representing an IRD life-cycle-phase shall have its SYSTEM-LOCK meta-attribute equal to ON if either of the following is satisfied:

- (1) It is either the archived or controlled IRD life-cycle-phase.
- (2) There exists at least one IRD entity which exists in the life-cycle-phase defined by the meta-entity.

Thus, in these cases, an IRD life-cycle-phase cannot be deleted from the IRD Schema.

Meta-relationships of type IRD-PARTITION-GREATER-THAN-IRD-PARTITION shall have the SYSTEM-LOCK meta-attribute equal to ON, if any one of the following is satisfied:

- (1) The meta-relationship associates the archived IRD life-cycle-phase to the controlled IRD life-cycle-phase.
- (2) There is at least one entity in the IRD life-cycle-phase defined by either partition meta-entity of the meta-relationship.
- (3) There is at least one entity in any IRD life-cycle-phase which is subordinate to the IRD life-cycle-phase defined by the second meta-entity of the meta-relationship.

The SYSTEM-LOCK meta-attribute equal to ON prevents any such meta-relationship from being deleted from the IRD Schema.

#### 4.2.1.3 Implications on IRD Maintenance

A meta-entity of type IRD-PARTITION which represents an IRD life-cycle-phase shall become effective only when it is incorporated into the Life-Cycle-Phase Hierarchy. Thus, an installation can construct a tree of uncontrolled IRD life-cycle-phases using IRD Schema commands or panels, and make all the IRD life-cycle-phases effective by adding only one meta-relationship. When an IRD life-cycle-phase becomes effective, entities can exist in it.

#### 4.2.2 IRD-Schema-Structures

An IRD-schema-structure is a set of entity-types and relationships-types such that if a given relationship-type is in the IRD-schema-structure, its associated entity-types are also in the IRD-schema-structure.

##### 4.2.2.1 Implementation in the IRD Schema

Each IRD-schema-structure is defined by a meta-entity of type IRD-SCHEMA-STRUCTURE; there can be more than one of these meta-entities with the same assigned-access-name. The modify-meta-entity-life-cycle-phase command applies to entities of type IRD-SCHEMA-STRUCTURE.

The membership of an IRD-schema-structure is defined by meta-relationships of the following types:

IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE  
 IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE

Both of these meta-relationship-types belong to the meta-relationship-class-type SET-OF, and have the following characteristics (see Subsection 9.2 of Module 1 of this standard):

- (1) They shall have a ratio of (0,n:0,m).
- (2) Only one such meta-relationship shall exist between a given pair of meta-entities.
- (3) They shall be nonsequenced.
- (4) The only meta-attribute-types which shall be associated with these meta-relationship-types are SYSTEM-LOCK and IMPLEMENTATION-LOCK.

If  $S$  is a meta-entity of type IRD-SCHEMA-STRUCTURE, and  $M$  is a meta-entity of type RELATIONSHIP-TYPE or ENTITY-TYPE such that the meta-relationship  $S$  SET-OF  $M$  exists, then  $M$  is called a member of the IRD-schema-structure  $S$ .

#### 4.2.2.2 IRD-Schema-Structure Subsets

The meta-relationship-type IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE is used to define substructures of a given IRD-schema-structure. This meta-relationship-type belongs to the meta-relationship-class-type of HAS-SUBSET. Its characteristics shall be as follows:

- (1) It has a ratio of (0,n:0,m).
- (2) It is unique with respect to meta-entity pairs.
- (3) It is nonsequenced.
- (4) The only meta-attribute-types associated with this meta-relationship-type shall be SYSTEM-LOCK and IMPLEMENTATION-LOCK.

If  $S$  and  $S'$  are meta-entities of type IRD-SCHEMA-STRUCTURE such that the meta-relationship  $S$  HAS-SUBSET  $S'$  exists, then  $S'$  is said to be a direct subset of the IRD-schema-structure  $S$ . Conversely,  $S$  is said to be a direct superset of the IRD-schema-structure  $S'$ . If  $S'$  is not a direct subset of  $S$ , but there is a set of one or more IRD-schema-structures  $S_1 \dots S_n$  such that the meta-relationships

$$S \text{ HAS-SUBSET } S_1 \dots S_i \text{ HAS-SUBSET } S_{i+1} \dots S_n \text{ HAS-SUBSET } S'$$

exist,  $S'$  is called an indirect subset of  $S$ . Conversely,  $S$  is called an indirect superset of  $S'$ . The term subset and superset apply to both direct and indirect subsets and supersets respectively.

#### 4.2.2.3 Integrity Constraints for IRD Schema Structures

In the definition of integrity constraints for IRD-schema-structures, the following notation is used:

Let  $S, S'$  denote meta-entities of type IRD-SCHEMA-STRUCTURE.  
 Let  $E$  denote a meta-entity of type ENTITY-TYPE.  
 Let  $R$  denote a meta-entity of type RELATIONSHIP-TYPE.  
 Let  $M$  denote a meta-entity of type ENTITY-TYPE or RELATIONSHIP-TYPE.

IRD-schema-structures shall be subject to the following integrity constraints:

- (1) The Completeness Constraint. If  $R$  is a member of  $S$  such that the meta-relationship  $R$  CONNECTS  $E$  exists, then  $E$  shall be a member of  $S$ .
- (2) The Subset-Membership Constraint. If  $M$  is a member of  $S'$ , and  $S'$  is a subset of  $S$ , then  $M$  shall be a member of  $S$ .
- (3) The Acyclic Nesting Constraint. Given an IRD-schema-structure  $S$ , both of the following shall be satisfied:
  - (a) The meta-relationship  $S$  HAS-SUBSET  $S$  cannot exist.

- (b) There shall be no IRD-schema-structure  $S'$  such that  $S'$  is a subset of  $S$  and  $S$  is a subset of  $S'$ .

In order to ensure that the above constraints are satisfied, the following IRD-schema-maintenance commands shall not operate on meta-entities of type IRD-SCHEMA-STRUCTURE:

```
add-meta-entity-command
modify-meta-entity-command
delete-meta-entity-command
copy-meta-entity-command
```

Furthermore the following commands shall not operate on any meta-relationship of type IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE, IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE, or IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE:

```
add-meta-relationship
modify-meta-relationship
delete-meta-relationship
```

The IRD Schema life-cycle-phase integrity constraint (defined in Subsection 9.2.4 of the standard IRDS, Module 1) applies to all meta-relationships involving IRD-schema-structures. The single controlled meta-entity version constraint (also defined in Subsection 9.8 of Module 1 of the standard IRDS) also applies to IRD-schema-structures.

#### 4.2.3 The Phase Sensitivity Structure

When the IRD is activated with this Module in effect, one IRD-schema-structure with the assigned-access-name PHASE-SENSITIVITY-STRUCTURE shall exist in the CONTROLLED IRD Schema life-cycle-phase (see Subsection 10.5). This meta-entity is called the Phase Sensitivity Structure. Other meta-entities with the same assigned-access-name can exist in other IRD Schema life-cycle-phases.

The Phase Sensitivity Structure defines the following:

- (1) The entity-types whose instances exist in IRD life-cycle-phases. Entities of these types can have user-designated version identifiers. An entity whose type is not in this IRD-schema-structure (e.g., a security entity) does not have a user designated version identifier. The modify entity life-cycle-phase command or panel applies to entities of each type in this IRD-schema-structure.
- (2) The relationship-types for which the IRD Life-Cycle-Phase Integrity Constraint applies (see Subsection 4.2.4). It is possible for this IRD-schema-structure to have no relationship-types; in this case, no relationship need conform to the IRD Life-Cycle-Phase Integrity Constraint.

The Phase Sensitivity Structure is itself subject to a special constraint, called the acyclic relationship-types constraint. This is defined as follows:

Let  $R_1 \dots R_n$  denote all the meta-entities of type RELATIONSHIP-TYPE which are members of an IRD-schema-structure  $S$ . Let  $E_1 \dots E_m$  denote all the distinct meta-entities of type ENTITY-TYPE such that for each  $E_i$ ,  $1 \leq i \leq m$ , there exists a  $R_j$  such that the meta-relationship  $R_j$  CONNECTS  $E_i$  exists.

The IRD-schema-structure  $S$  is said to be acyclic with respect to relationship-types if and only if there is no subset of relationship-types  $R_j$ ,  $1 \leq j \leq n$ , such that the following meta-relationships exist:

$R_{j1}$  CONNECTS  $E_{i1}$  POSITION = 1  
 $R_{j1}$  CONNECTS  $E_{i2}$  POSITION = 2  
  
 $R_{j2}$  CONNECTS  $E_{i2}$  POSITION = 1  
 $R_{j2}$  CONNECTS  $E_{i3}$  POSITION = 2  
  
 ...  
  
 $R_{jk}$  CONNECTS  $E_{ik}$  POSITION = 1  
 $R_{jk}$  CONNECTS  $E_{i1}$  POSITION = 2

(Here each  $E_{ix}$ ,  $1 \leq i \leq jk$ ,  $1 \leq x \leq k$  is a distinct meta-entity of type ENTITY-TYPE which is a member of  $S$ .)

Note that it is possible for  $S$  to have a relationship-type  $R_j$  such that

$R_j$  CONNECTS  $E_i$  POSITION = 1  
 $R_j$  CONNECTS  $E_i$  POSITION = 2

In other words, an acyclic IRD-schema-structure can contain recursive relationship-types.

The acyclic relationship-types constraint states that a given IRD-schema-structure shall be acyclic with respect to nonrecursive relationship-types.

NOTE: An implementor may enforce this constraint in one of at least two ways:

- (1) It may be enforced whenever a meta-entity with the assigned-access-name, PHASE-SENSITIVITY-STRUCTURE, is maintained.
- (2) It may be enforced when a meta-entity with assigned-access-name PHASE-SENSITIVITY-STRUCTURE is moved into the CONTROLLED IRD-schema-life-cycle-phase.

#### 4.2.4 IRD Life-Cycle-Phase Integrity

IRD Life-cycle-phase Integrity specifies how entities in different life-cycle-phases can be related. It is dependent both on the Life Cycle Phase Hierarchy and the Phase Sensitivity Structure. The following definitions and rules define life-cycle-phase integrity:

- Definition 1. A given relationship-type  $R$  is phase-related if it is a member of the Phase Sensitivity Structure. Note that  $R$  designates the forward relationship-type.
- Definition 2. A given forward relationship  $E1 R E2$  is phase-related if the relationship-type of which it is an instance is phase-related.
- Definition 3. Entity  $E1$  is said to depend on entity  $E2$  if and only if there exists a forward relationship  $E1 R E2$  which is phase-related.
- Rule 1. If  $E1$  is in an uncontrolled IRD life-cycle-phase, then  $E1$  shall only depend on entities in any uncontrolled IRD life-cycle-phase or the controlled IRD life-cycle-phase.



- Rule 2. If *E1* is in the archived IRD life-cycle-phase, then *E1* shall only depend on entities in the controlled IRD life-cycle-phase or the archived IRD life-cycle-phase.
- Rule 3. If *E1* is in the controlled IRD life-cycle-phase, then *E1* shall only depend on entities in the controlled IRD life-cycle-phase.

#### 4.2.5 The IRD-Schema-Structure IRD-STRUCTURE

When the IRD is activated with this Module in effect, there shall be one meta-entity of type IRD-SCHEMA-STRUCTURE in the CONTROLLED IRD Schema life-cycle-phase (See Subsection 10.5). This meta-entity, called the IRD Structure, defines all entity-types and relationship-types whose instances can exist in the IRD.

This meta-entity shall be subject to the constraint that each member shall have a distinct assigned-access-name.

The above replaces the single controlled meta-entity version constraint, which is defined in Subsection 9.8 of Module 1.

The Phase Sensitivity Structure shall always be a subset of the IRD Structure. This structure contains all those entity-types contained in the IRD Structure, instances of which can exist in IRD life-cycle-phases, and those relationship-types which the IRD Life-Cycle-Phase Constraint applies.

## 5 Basic Language Elements

### 5.1 Function

To identify those additional terminal and non-terminal symbols which are:

- (1) Provided with this Module of the standard IRDS, and
- (2) Not defined in any other Module upon which this Module depends.

### 5.2 Format

```

append-word ::= APPEND
disconnect-word ::= DISCONNECT
include-word ::= INCLUDE
install-word ::= INSTALL
members-word ::= MEMBERS | MEMBER
subsets-word ::= SUBSETS | SUBSET
IRD-schema-structure-word ::= IRD-SCHEMA-STRUCTURE

```

```

meta-entity-access-name-list ::=
    meta-entity-access-name M1.4.4
    [ , meta-entity-access-name-list ]

```

```
/* Additional irds-reserved-words */
irds-module-4-reserved-words :=
    append-word
    disconnect-word
    include-word
```

### 5.3 Syntax Rules

- (1) The singular and plural forms of a word are provided to enhance readability. In such cases, the singular and plural forms of the word are equivalent.
- (2) The implementation may allow for abbreviated forms of irds-words.

**5.4 General Rules** None.

## 6 IRD Schema Structure Manipulation Commands

**Function** The following commands maintain IRD-schema-structures.

### Format

```
/* None. This breakdown is explanatory only. */
```

```
IRD-schema-structure-command ::=
    add-IRD-schema-structure-command           6.1
  | modify-IRD-schema-structure-command       6.2
  | delete-IRD-schema-structure-command       6.3
  | copy-IRD-schema-structure-command         6.4
  | install-IRD-structure-command            6.5
```

**Syntax Rules** None.

**General Rules** None.

### Access Rules

The following rules specify access constraints for the above commands. For the Access Rules which follow:

Let  $U$  denote the effective IRDS-user.  
Let  $V$  denote the effective IRD-schema-view.  
Let  $V_1 \dots V_n$  denote IRD-schema-views which are related to  $U$ .

- (1) For the add-IRD-schema-structure command, modify-IRD-schema-structure command and copy-IRD-schema-structure command,  $V$  shall have an IRD-SCHEMA-PHASE-NAME attribute equal to UNCONTROLLED.
- (2) For the install-IRD-structure command,  $V$  shall have an IRD-SCHEMA-PHASE-NAME attribute equal to CONTROLLED.

- (3) For the modify-IRD-schema-structure command, and delete-IRD-schema-structure command, the IRD-schema-structure specified in the command shall be in the IRD Schema life-cycle-phase identified by the IRD-SCHEMA-PHASE-NAME attribute of V.
- (4) For the copy-IRD-schema-structure command and the install-IRD-structure command, there shall be a  $V_i$ ,  $1 \leq i \leq n$ , such that the IRD-SCHEMA-PHASE-NAME attribute of  $V_i$  identifies the IRD Schema life-cycle-phase which contains the existing IRD-schema-structure specified in the command.

### Security Rules

- (1) If the facilities of Module 3, "IRDS Security", are in effect, all Security Rules which apply to all IRD Schema commands and IRD Schema maintenance commands as defined in Module 3 of the standard IRDS apply to these commands.
- (2) All Security Rules specified for each command presume that the facilities of Module 3 are in effect. If the facilities of Module 3 are not in effect, then the Security Rules specified for each command shall be ignored.

**Actions** None

### Error and Warning Conditions

- (1) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle. See Access Rules (1) and (2).
- (2) Error E01067: Meta-entity does not exist or is not visible. See Access Rules (3) and (4).

NOTE: Consult Section 8 of Module 3. This section describes how a user's permissions for the IRD Schema are defined. It also defines the terminology used in all the Security Rules in this Module.

## 6.1 Add IRD-Schema-Structure Command

**Function** To create an IRD-schema-structure.

### Format

```

add-IRD-schema-structure-command ::=
    add-word                               M1.4.3
    IRD-schema-structure-word              5
    new-meta-entity-access-name           M1.4.
    [ meta-entity-descriptive-name-clause ] M1.6.126
    [ new-meta-attributes-clause ]        M1.6.103
    [ append-clause ]                      7.1
    [ include-clause ]                     7.2
    ;

```

### Syntax Rules

- (1) The optional command-clauses may be specified in any order.

## General Rules

For the General Rules, Security Rules, and Actions which follow:

Let  $S$  denote the IRD-schema-structure identified by new-meta-entity-access-name.  
Let  $E1, E2$  denote meta-entities of type ENTITY-TYPE.  
Let  $R$  denote a meta-entity of type RELATIONSHIP-TYPE.  
Let  $S1 \dots Sk$  denote meta-entities of type IRD-SCHEMA-STRUCTURE.

- (1) IRD Schema life-cycle-phase integrity shall be preserved. That is:
  - (a)  $S$  shall be created in the UNCONTROLLED IRD Schema life-cycle-phase.
  - (b) If  $Sa, 1 \leq a \leq k$ , is specified in the subsets-clause subclause of the include-clause,  $Sa$  shall be in either the UNCONTROLLED or CONTROLLED IRD Schema life-cycle-phase.
  - (c) If  $Sa, 1 \leq a \leq k$  is specified in the append-clause, and  $Sa$  is in the ARCHIVED IRD Schema life-cycle-phase, each member of  $Sa$  shall be in the CONTROLLED IRD Schema life-cycle-phase.
  - (d) Each  $Ei, 1 \leq i \leq n$ , or  $Rj, 1 \leq j \leq m$ , specified in the members-clause subclause of the include-clause, shall be in either the UNCONTROLLED or CONTROLLED IRD Schema life-cycle-phase.
- (2) Assume  $R$  is associated with meta-entities  $E1, E2$ . If  $R$  is specified in the members-clause subclause of the include-clause, then  $E1$  and  $E2$  shall each be:
  - (a) specified in the members-clause subclause of include-clause, or
  - (b) a member of an IRD-schema-structure specified in the subsets-clause subclause of the include-clause, or
  - (c) a member of an IRD-schema-structure specified in the append-clause.
- (3) The results of the append-clause shall be evaluated prior to evaluating the include-clause. Within the include-clause, the results of the subsets-clause shall be evaluated before the members-clause.

## Security Rules

For the following Security Rules:

Let  $U$  be the effective IRDS-user.  
Let  $V$  be the effective IRD-schema-view.  
Let  $V1$  be another IRD-schema-view related to  $U$ .

- (1)  $V$  shall authorize ADD-PERMISSION for meta-entities of type IRD-SCHEMA-STRUCTURE. Consult the Security Rules in Section 8 of Module 3.
- (2)  $U$  shall be related to one or two IRD-schema-views,  $V [V1]$  which authorize UPDATE-PERMISSION for meta-relationships of each of the following types:

IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE  
 IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE  
 IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE.

- (a) In order to establish meta-relationships which have both meta-entities in the UNCONTROLLED IRD Schema life-cycle-phase, *V* shall authorize UPDATE-PERMISSION for meta-relationships of the types listed above.
- (b) In order to establish meta-relationships which have the first meta-entity is in the UNCONTROLLED IRD Schema life-cycle-phase and the second meta-entity is in the CONTROLLED IRD Schema life-cycle-phase, *V* and *V1* shall authorize UPDATE-PERMISSION for meta-relationships of the types listed above. (Note that *V1* shall be associated with the CONTROLLED IRD Schema life-cycle-phase.)

### Actions

- (1) The command shall be validated. If any errors are encountered, all error and warning messages shall be produced. No update of the IRD Schema takes effect.
- (2) If no errors were encountered, then:
  - (a) A meta-entity of type IRD-SCHEMA-STRUCTURE with the specified access-name shall be added to the IRD Schema.
  - (b) The following meta-attributes shall be created with the following values:

ADDED-BY = name of effective IRDS user  
 NUMBER-OF-TIMES-MODIFIED = 0

The meta-attribute-group DATE-TIME-ADDED shall be created. Within this meta-attribute-group:

SYSTEM-DATE = date of transaction  
 SYSTEM-TIME = time of transaction.

- (c) All meta-attributes specified in new-meta-attributes-clause shall be created for the IRD-schema-structure *S*.
  - (d) Let *S<sub>a</sub>* denote an IRD-schema-structure specified in the append-clause. Then each member of *S<sub>a</sub>* becomes a member of *S*.
  - (e) Let *S<sub>s</sub>* denote an IRD-schema-structure specified in the subsets-clause subclause of the include-clause. Then each member of *S<sub>s</sub>* becomes a member of *S* and *S<sub>s</sub>* becomes a direct subset of *S*.
  - (f) A meta-relationship shall be created from the new IRD-schema-structure to each entity-type and relationship-type meta-entity specified in the member-clause subclause of the include-clause.
- (3) Completion of command processing shall be confirmed.

**Error and Warning Conditions**

- (1) Error E01087: Creation of meta-relationship would violate IRD Schema life-cycle-phase integrity. See General Rule (1).
- (2) Error E04001: Participating entity-type for a relationship-type not specified. See General Rule (2).
- (3) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (1).
- (4) Error E03004: User not authorized to perform specified action(s). See Security Rule (2).

**6.2 Modify IRD-Schema-Structure Command**

**Function** To modify an IRD-schema-structure or create a new IRD-schema-structure with the same assigned-access-name as the specified IRD-schema-structure.

**Format**

```

modify-IRD-schema-structure-command ::=
    modify-word                M1.4.3
    IRD-schema-structure-word  5
    existing-meta-entity-access-name M1.4.4
    [ new-meta-entity-version-clause ] M1.6.28
    [ meta-entity-descriptive-name-clause ] M1.6.126
    [ modified-meta-attributes-clause ] M1.6.104
    [ disconnect-clause ] 7.3
    [ exclude-clause ] 7.4
    [ append-clause ] 7.1
    [ include-clause ] 7.2
    ;
    
```

**Syntax Rules**

- (1) The optional command-clauses may be specified in any order.

**General Rules**

For the General Rules, Security Rules and Actions which follow:

- Let *S* denote the meta-entity identified by existing-meta-entity-access-name.
- Let *P* denote the IRD Schema life-cycle-phase in which *S* exists.
- Let *M* denote a member of *S*.
- Let *S'* denote a direct subset of *S*.
- Let *R* denote a meta-entity of type RELATIONSHIP-TYPE.
- Let *E* denote a meta-entity of type ENTITY-TYPE.

- (1) Existing-meta-entity-access-name shall identify a meta-entity of type IRD-SCHEMA-STRUCTURE.
- (2) If *S* is in the CONTROLLED or ARCHIVED IRD Schema life-cycle-phase, then new-meta-entity-version-clause shall be required. As a consequence, if new-meta-entity-version-

clause is not specified, *S* shall be in the UNCONTROLLED IRD Schema life-cycle-phase.

- (3) Meta-entity-descriptive-name-clause shall be allowed only if *S* has no descriptive-name.
- (4) The following command-clauses shall be evaluated in the order given:
  - (a) disconnect-clause
  - (b) exclude-clause
  - (c) append-clause
  - (d) include-clause

Within the exclude-clause, the members-clause shall be evaluated before the subsets-clause. Within the include-clause, the subsets-clause shall be evaluated before the members-clause.

- (5) Any IRD-schema-structure specified in disconnect-clause or exclude-clause shall be a direct subset of the IRD-schema-structure being modified.
- (6) No IRD-schema-structure specified in the disconnect-clause shall be specified in the exclude-clause. (This is a consequence of General Rules (4) and (5).)
- (7) Each meta-entity specified in the members-clause subclause of the exclude-clause shall be a member of *S*.
- (8) If *M* is specified in the members-clause subclause of the exclude-clause, then either:
  - (a) *M* shall not be a member of any *S'*, or
  - (b) Each *S'* which contains *M* as a member shall be either:
    - (i) Specified in the disconnect-clause, or
    - (ii) Specified in the subsets-clause subclause of exclude-clause.

NOTE: This test is sufficient because each member of an indirect subset of *S* is also a member of a direct subset of *S*.

- (9) If *E* and *R* are members of *S* such that there exists a meta-relationship of type RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE which associates *R* and *E*, then if *E* is specified in the exclude-clause, *R* shall also be specified in the exclude-clause.
- (10) IRD Schema life-cycle-phase integrity shall be preserved. That is:
  - (a) If *S* is in the ARCHIVED IRD Schema life-cycle-phase, each member of *S* which is in the ARCHIVED IRD Schema life-cycle-phase shall be specified in the members-clause subclause of the exclude-clause.
  - (b) If *S* is in the ARCHIVED IRD Schema life-cycle-phase, each direct subset of *S* which is in the ARCHIVED IRD Schema life-cycle-phase shall be specified in the subsets-clause subclause of the exclude-clause.
  - (c) No meta-entity in the ARCHIVED IRD Schema life-cycle-phase shall be specified in the include-clause.

- (d) No IRD-schema-structure in the ARCHIVED IRD Schema life-cycle-phase shall be specified in the append-clause, unless each of its members is in the CONTROLLED IRD Schema life-cycle-phase.

(11) If both of the following are satisfied:

- (a)  $R$  and  $E$  are such that there exists a meta-relationship of type RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE which associates  $R$  and  $E$ .
- (b)  $R$  is specified in the members-clause subclause of the include-clause.

then:

- (i)  $E$  shall be a member of the IRD-schema-structure being maintained after the exclude-clause is evaluated, or
- (ii)  $E$  shall be specified in the members-clause subclause of include-clause, or
- (iii)  $E$  shall be a member of an IRD-schema-structure specified in append-clause, or
- (iv)  $E$  shall be a member of an IRD-schema-structure specified in the subsets-clause subclause of include-clause.

(12) The acyclic nesting constraint shall not be violated. That is:

- (a)  $S$  shall not be specified in the include-clause.
- (b) No (direct or indirect) superset of  $S$  shall be specified in the include-clause.

(13) A warning shall be produced if either  $S$  or any superset of  $S$  is specified in the append-clause.

### Security Rules

For the following Security Rules:

Let  $U$  denote the effective IRDS-user.  
Let  $V$  denote the effective IRD-schema-view.  
Let  $V1, V2$  denote IRD-schema-views related to  $U$ .

- (1) If new-meta-entity-version-clause is not specified then  $V$  shall authorize MODIFY-PERMISSION for meta-entities of type IRD-SCHEMA-STRUCTURE.
- (2) If new-meta-entity-version-clause is specified, then:
  - (a)  $V$  shall authorize ADD-PERMISSION for meta-entities of type IRD-SCHEMA-STRUCTURE.
  - (b) There shall be a  $V1$  which authorizes READ-PERMISSION to meta-entities of type IRD-SCHEMA-STRUCTURE in  $P$ .
- (3) Each meta-relationship of one of the following types:



IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE  
 IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE  
 IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE

in which *S* participates shall be completely visible to *U*. In particular:

- (a) For each such meta-relationship in which both meta-entities are in the same IRD Schema life-cycle-phase, there shall be a *V1* in which the meta-relationship is completely visible.
  - (b) For each meta-relationship in which the participating meta-entities are in different IRD Schema life-cycle-phases, there shall exist *V1* and *V2* in which the meta-relationship is completely visible.
- (4) *U* shall be related to appropriate IRD-schema-views which authorize the creation of new meta-relationships and the deletion of existing meta-relationships of the types identified in Security Rule (3). In particular:
- (a) In order to maintain meta-relationships which have both meta-entities in the UNCONTROLLED IRD Schema life-cycle-phase, *V* shall authorize UPDATE-PERMISSION for meta-relationships of the types listed above.
  - (b) In order to maintain meta-relationships which have the first meta-entity is in the UNCONTROLLED IRD Schema life-cycle-phase and the second meta-entity is in the CONTROLLED IRD Schema life-cycle-phase, *V* and *V1* shall authorize ADD-PERMISSION for meta-relationships of the types listed above. (Note that *V1* shall be associated with the CONTROLLED IRD Schema life-cycle-phase.)
- (5) If new-meta-entity-version-clause is specified, then conformance to Security Rules (2) and (3) shall be necessary and sufficient to authorize that any member or subset be excluded or that any subset be disconnected.

### Actions

- (1) Validation shall proceed as follows:
- (a) Syntax validation and the validation specified in General Rules (1) through (3) is performed, and conformance with the Security Rules shall be determined.
- The order in which these actions are performed shall be implementation-dependent.
- (b) The effects of the command on the membership and subsets of the maintained IRD-schema-structure shall be evaluated in the order specified in General Rule (4). Validation of the command according to General Rules (5) through (13) shall be performed within the sequence identified in General Rule (4).

If any errors are encountered, all appropriate error and warning messages shall be produced. No update of the IRD Schema shall be performed.

- (2) If no errors are encountered and new-meta-entity-version-clause is not specified, then all modifications to attributes, membership and subsetting of the modified IRD-schema-structure shall be performed. The audit meta-attributes of the maintained IRD-schema-structure shall be updated as follows:

MODIFIED-BY = name of the effective IRDS user  
NUMBER-OF-TIMES-MODIFIED = current value + 1

The component meta-attributes of the meta-attribute-group DATE-TIME-MODIFIED shall be updated as follows:

SYSTEM-DATE = date of transaction  
SYSTEM-TIME = time of transaction

All additions to the membership of *S* shall also be applied to all direct and indirect supersets of *S*. The audit meta-attributes of these IRD-schema-structures shall be modified as are those of *S*.

If meta-entity-descriptive-name-clause is specified, then all meta-entities with the same assigned-access-name as *S* shall be given the specified descriptive-name. The audit meta-attributes of these IRD-schema-structures shall be modified as are those of *S*.

All appropriate warning messages shall be produced.

- (3) If no errors are encountered and new-meta-entity-version-clause is specified, then a new IRD-schema-structure shall be created.

All user-specifiable meta-attributes not specified in modified-meta-attributes-clause will be the same as those for *S*. The meta-attributes specified in the modified-meta-attribute-clause will be as specified in that command-clause.

The meta-relationships from the new IRD-schema-structure to other meta-entities resulting from this command shall be the same as if the new IRD-schema-structure were created via a copy-IRD-schema-structure command, and subsequently modified by the disconnect-clause, exclude-clause, include-clause and append-clause specified in the modify-IRD-schema-structure command.

The audit meta-attributes of the new IRD-schema-structure shall be as created by an add-IRD-schema-structure-command.

If meta-entity-descriptive-name-clause is specified, then all meta-entities with the same assigned-access-name as *S* shall be given the specified descriptive-name. The audit meta-attributes of these IRD-schema-structures shall be modified as specified in Action (2) above.

- (4) Completion of command processing shall be confirmed.

#### **Error and Warning Conditions**

- (1) Error E04002: Specified meta-entity is not an IRD-schema-structure. See General Rule (1).
- (2) Error E01076: Cannot modify specified meta-entity. New version is required. See General Rule (2).
- (3) Error E01075: Meta-entity already has a descriptive-name. See General Rule (3).
- (4) Error E04003: IRD-schema-structure in command-clause is not a direct subset of IRD-schema-structure being maintained. See General Rules (5) and (6).

- (5) Error E04004: Meta-entity specified in exclude-clause is not a member of IRD-schema-structure being maintained. See General Rule (7).
- (6) Error E04005: Specified member is in a subset of maintained IRD-schema-structure and cannot be excluded. See General Rule (8).
- (7) Error E04006: Specified entity-type participates in a relationship-type which is a member of maintained IRD-schema-structure. See General Rule (9).
- (8) Error E01087: Creation of meta-relationship would violate IRD Schema life-cycle-phase integrity. See General Rule (10).
- (9) Error E04001: Participating entity-type for a relationship-type not specified. See General Rule (11).
- (10) Error E04007: Command would create a cycle of IRD-schema-structures. See General Rule (12).
- (11) Warning W04001: Command calls for appending a superset of specified IRD-schema-structure.
- (12) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rules (1) and (2)(a).
- (13) Error E03004: User not authorized to perform specified action(s). See Security Rules (2)(b), (3), (4), and (5).

### 6.3 Delete IRD-Schema-Structure Command

**Function** To delete an IRD-schema-structure.

**Format**

```
delete-IRD-schema-structure-command ::=
    delete-word                M1.4.3
    IRD-schema-structure-word  5
    existing-meta-entity-access-name M1.4.4
    ;
```

**Syntax Rules** None

**General Rules**

For the General Rules, Security Rules and Actions which follow:

Let *S* denote the meta-entity identified by existing-meta-entity-access-name.  
 Let *P* denote the IRD Schema life-cycle-phase in which *S* exists.  
 Let *S'*, *S''* denote meta-entities of type IRD-SCHEMA-STRUCTURE.  
 Let *M* denote a meta-entity of type ENTITY-TYPE or RELATIONSHIP-TYPE.

- (1) *S* shall be a meta-entity of type IRD-SCHEMA-STRUCTURE.
- (2) *S* shall not be a subset of any other IRD-schema-structure *S''*.

- (3) *S* shall not have either the SYSTEM-LOCK or IMPLEMENTATION-LOCK meta-attribute equal to ON.

### Security Rules

For the Security Rules which follow:

Let *U* denote the effective IRDS-user.

Let *V* denote the effective IRD-schema-view.

Let *P1* denote an IRD Schema life-cycle-phase not equal to *P*.

Let *V1* denote an IRD-schema-view related to *U* and associated with the IRD Schema life-cycle-phase *P1*.

- (1) *V* shall authorize DELETE-PERMISSION for meta-entities of type IRD-SCHEMA-STRUCTURE. See Module 3, Subsection 8.1.
- (2) *U* shall be related appropriate IRD-schema-views which authorize the deletion of all meta-relationships of the following types

IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE

IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE

IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE

in which *S* participates. In particular:

- (a) In order to delete meta-relationships which have both meta-entities in *P*, *V* shall authorize UPDATE-PERMISSION for meta-relationships of the types listed above.
- (b) In order to delete meta-relationships which have the first meta-entity in *P* and the second meta-entity in *P1*, *V* and *V1* shall authorize UPDATE-PERMISSION for meta-relationships of the types listed above.

### Actions

- (1) The command shall be validated. If any error is encountered, all appropriate error messages shall be produced. No change of the IRD Schema takes place.
- (2) If no errors were encountered, then:
  - (a) For each *S'*, a direct subset of *S*, the meta-relationship *S* HAS-SUBSET *S'* shall be deleted.
  - (b) For each *M*, a member of *S*, the meta-relationship *S* SET-OF *M* shall be deleted.
  - (c) The meta-entity *S* shall be deleted from the IRD Schema.
- (3) Completion of command processing shall be confirmed.

### Error and Warning Conditions

- (1) Error E04002: Specified meta-entity is not an IRD-schema-structure. See General Rule (1).

- (2) Error E01079: Cannot delete specified meta-entity. Other meta-entities depend on specified meta-entity. See General Rule (2).
- (3) Error E01073: Meta-entity is locked and cannot be deleted. See General Rule (3).
- (4) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (1).
- (5) Error E03004: User not authorized to perform specified action(s). See Security Rule (2).

#### 6.4 Copy IRD-Schema-Structure Command

**Function** To create a new-IRD-schema-structure with the same meta-attributes, members and subsets as an existing IRD-schema-structure.

#### Format

```

copy-IRD-schema-structure-command ::=
    copy-word                M1.4.3
    IRD-schema-structure-word 5
    existing-meta-entity-access-name M1.4.4
    to-word                  M1.4.3
    new-IRD-schema-structure
    [ meta-entity-descriptive-name-clause ] M1.6.126
    ;

new-IRD-schema-structure ::=
    new-meta-entity-access-name M1.4.4
    | new-meta-entity-version-clause M1.6.28

```

#### Syntax Rules

- (1) If new-meta-entity-access-name is specified, the assigned-access-name component shall not be the same as the assigned access-name component of existing meta-entity-access-name.

#### General Rules

For the General Rules, Security Rules and Actions which follow:

Let *S* be the meta-entity identified by existing-meta-entity-access-name.  
 Let *P* be the IRD Schema life-cycle-phase in which *S* exists.  
 Let *S'* be a direct subset of *S*.  
 Let *S''* be the meta-entity identified by new-IRD-schema-structure.  
 Let *M* denote a member of *S*.

- (1) Existing-meta-entity-access-name shall identify a meta-entity of type IRD-SCHEMA-STRUCTURE.
- (2) *S''* shall be created in the UNCONTROLLED IRD Schema life-cycle-phase. That is, the effective IRD-schema-view shall have the IRD-SCHEMA-PHASE-NAME attribute equal to UNCONTROLLED.

- (3) If *S* is in the ARCHIVED IRD Schema life-cycle-phase, then each member and direct subset of *S* shall be in the CONTROLLED IRD Schema life-cycle-phase.
- (4) If new-meta-entity-version-clause is specified, and *S* has a descriptive-name, then meta-entity-descriptive-name-clause shall not be allowed.

### Security Rules

For the Security Rules which follow:

Let *U* denote the effective IRDS-user.  
Let *V* denote the effective IRD-schema-view.

- (1) *V* shall authorize ADD-PERMISSION for meta-entities of type IRD-SCHEMA-STRUCTURE.

(NOTE: *V* is associated with the UNCONTROLLED IRD Schema life-cycle-phase.)

- (2) If *P* is not equal to UNCONTROLLED, then there shall be an IRD-schema-view *V1* for which the following are satisfied:
  - (a) *V1* is related to *U*.
  - (b) *V1* is associated with *P*.
  - (c) *V1* authorizes READ-PERMISSION for meta-entities of type IRD-SCHEMA-STRUCTURE.
- (3) Each meta-relationship of one of the following types

IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE  
IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE  
IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE

in which *S* participates shall be completely visible to *U*. In particular:

- (a) For each such meta-relationship in which both meta-entities are in the same IRD Schema life-cycle-phase, there shall be a *V1* in which the meta-relationship is completely visible.
  - (b) For each meta-relationship in which the participating meta-entities are in different IRD Schema life-cycle-phases, there shall exist *V1* and *V2* in which the meta-relationship is completely visible.
- (4) *U* shall be related to appropriate IRD-schema-views which authorize the creation of new meta-relationships and the deletion of existing meta-relationships of the types identified in Security Rule (3). In particular:
    - (a) In order to maintain meta-relationships which have both meta-entities in the UNCONTROLLED IRD Schema life-cycle-phase, *V* shall authorize UPDATE-PERMISSION for meta-relationships of the types listed above.
    - (b) In order to maintain meta-relationships which have the first meta-entity is in the UNCONTROLLED IRD Schema life-cycle-phase and the second meta-entity is in the CONTROLLED IRD Schema life-cycle-phase, *V* and *V1* shall authorize ADD-

PERMISSION for meta-relationships of the types listed above. (Note that *VI* shall be associated with the CONTROLLED IRD Schema life-cycle-phase.)

### Actions

- (1) The command shall be validated. If any error is encountered, all error messages shall be produced. No change shall take place in the IRD Schema.
- (2) If no errors are encountered, then a new meta-entity, *S''*, shall be created in the UNCONTROLLED IRD-schema-structure. *S''* shall be given the same user-specifiable meta-attributes as *S*. The audit meta-attributes of *S''* shall be as specified in the add-meta-entity-command.
  - (a) *S* and *S''* have identical members and subsets. That is:
    - (i) For each *M* which is a member of *S*, the meta-relationship *S''* SET-OF *M* shall be created.
    - (ii) For each *S'* which is a direct subset of *S*, the meta-relationship *S''* HAS-SUBSET *S'* shall be created.
  - (b) If *S* and *S''* have distinct assigned-access-names, *S''* shall be given the descriptive-name as specified in this command.
  - (c) If *S* and *S''*:
    - (i) Have the same assigned-access-name, and
    - (ii) *S* does not have a descriptive-name, and
    - (iii) meta-entity-descriptive-name-clause is specified,
 then:
    - (i) *S''* shall be given the specified descriptive-name.
    - (ii) Each meta-entity with the same assigned-access-name as *S* shall be given the specified descriptive-name. The audit meta-attributes of each of these meta-entities shall be modified as specified in the modify-meta-entity-command.
- (3) Completion of command processing shall be confirmed.

### Error and Warning Conditions

- (1) Error E04002: Specified meta-entity is not an IRD-schema-structure. See General Rule (1).
- (2) Error E01063: Effective IRD-schema-view not associated with correct IRD Schema life-cycle. See General Rule (2).
- (3) Error E01087: Creation of meta-relationship would violate IRD Schema life-cycle-phase integrity. See General Rule (3).
- (4) Error E01075: Meta-entity already has a descriptive-name. See General Rule (4).

## 6.5 Install IRD-Structure Command

**Function** To move one set of meta-entities into the CONTROLLED IRD Schema life-cycle-phase, and to move superseded meta-entities into the ARCHIVED IRD Schema life-cycle-phase.

### Format

```
install-IRD-structure-command ::=
    install-word                    5
    existing-meta-entity-access-name M1.4.4
    ;
```

**Syntax Rules** None

### General Rules

The following definitions are required for the following rules and actions:

Let *E1* and *E2* be meta-entities with the same assigned-access-name. Assume that *E1* is in the CONTROLLED IRD Schema life-cycle-phase. *E2* is said to supersede *E1* when *E2* is moved to the CONTROLLED IRD Schema life-cycle-phase. *E1* is said to be the superseded meta-entity, and *E2* is said to be the superseding meta-entity.

A meta-entity *C* is said to be a component of an IRD-schema-structure *S* if one of the following is satisfied:

- (a) *C* is a member of *S*
  - (b) *C* is a (direct or indirect) subset of *S*.
  - (c) *C* is a meta-entity of type RELATIONSHIP-CLASS-TYPE such that *C* is associated with a meta-entity of type RELATIONSHIP-TYPE which is a component of *S*.
  - (d) *C* is a meta-entity of type VARIATION-NAMES-DATA such that *C* is associated with a meta-entity of type ENTITY-TYPE which is a component of *S*.
  - (e) *C* is a meta-entity of type ATTRIBUTE-GROUP-TYPE such that *C* is associated with a meta-entity of type ENTITY-TYPE or RELATIONSHIP-TYPE which is a component of *S*.
  - (f) *C* is a meta-entity of type ATTRIBUTE-TYPE such that *C* is associated with a meta-entity of type ENTITY-TYPE, RELATIONSHIP-TYPE, or ATTRIBUTE-GROUP-TYPE which is a component of *S*.
  - (g) *C* is a meta-entity of type ATTRIBUTE-TYPE-VALIDATION-PROCEDURE or ATTRIBUTE-TYPE-VALIDATION-DATA which is associated with a meta-entity of type ATTRIBUTE-TYPE which is a component of *S*.
- (1) Existing-meta-entity-access-name shall identify a meta-entity of type IRD-SCHEMA-STRUCTURE.
  - (2) Existing-meta-entity-access-name shall have an assigned-access-name equal to IRD-STRUCTURE.



- (3) Existing-meta-entity-access-name shall identify a meta-entity in the UNCONTROLLED IRD Schema life-cycle-phase.
- (4) The IRD shall be in a deactivated state.
- (5) The IRDS-user who issues this command shall be the one who issued the deactivate-IRD command.
- (6) No two components of the specified IRD-schema-structure shall have the same assigned-access-name.
- (7) The specified IRD-schema-structure shall have a component meta-entity of type IRD-SCHEMA-STRUCTURE with the assigned-access-name PHASE-SENSITIVITY-STRUCTURE.

### Security Rules

For the Security Rules which follow:

Let  $U$  denote the effective IRDS-user.  
Let  $V$  denote the effective IRD-schema-view.

- (1)  $V$  shall authorize ADMINISTRATOR-PERMISSION.

NOTE: By the Access Rules in Section 6,  $V$  shall be associated with the CONTROLLED IRD Schema life-cycle-phase.

- (2)  $U$  shall be related to at least two other IRD-schema-views.
  - (a) One IRD-schema-view shall authorize ADMINISTRATOR-PERMISSION in the UNCONTROLLED IRD Schema life-cycle-phase.
  - (b) One IRD-schema-view shall authorize ADMINISTRATOR-PERMISSION in the ARCHIVED IRD Schema life-cycle-phase.

### Actions

- (1) The command shall be validated for conformance to the Security Rules and General Rules (1) through (5). The components of the specified IRD-schema-structure shall be identified, and conformance to General Rules (6) and (7) shall be determined. All appropriate error and warning messages shall be produced. If any errors are encountered, no changes shall be made to the IRD Schema.
- (2) If no errors were encountered, then the following actions shall be performed:
  - (a) The specified IRD-schema-structure and each of its components in the UNCONTROLLED IRD Schema life-cycle-phase shall be moved to the CONTROLLED IRD Schema life-cycle-phase.
  - (b) Any superseded meta-entities shall be moved to the ARCHIVED IRD Schema life-cycle-phase.
  - (c) The system-maintained meta-attributes of the meta-entities which moved across IRD Schema life-cycle-phases shall be modified in accordance with the rules of the modify-meta-entity-life-cycle-phase-command (Module 1, Subsection 5.1.1.9).

- (3) Completion of command processing shall be confirmed.

### Error and Warning Conditions

- (1) Error E04002: Specified meta-entity is not an IRD-schema-structure. See General Rule (1).
- (2) Error E04008: Assigned-access-name must be IRD-STRUCTURE. See General Rule (2).
- (3) Error E04009: Specified meta-entity not in UNCONTROLLED phase. See General Rule (3).
- (4) Error E01098: IRD must be deactivated. See General Rule (4).
- (5) Error E01099: Another user deactivated the IRD. See General Rule (5).
- (6) Error E04010: IRD-schema-structure has two distinct components with the same assigned-access-name. See General Rule (6).
- (7) Error E04019: Phase Sensitivity Structure not defined. See General Rule (7).
- (8) Error E03002: Effective IRD-schema-view does not authorize user to use command. See Security Rule (1).
- (9) Error E03004: User not authorized to perform specified action(s). See Security Rule (2).

## 7 Command-Clauses

The following command-clauses are used in the commands specified in this Module of the standard IRDS. These command-clauses have not been specified in previous Modules of this standard.

### 7.1 Append Command-Clause

**Function** When this command-clause is specified in add-IRD-schema-structure command or modify-IRD-schema-structure command, the members of the IRD-schema-structures specified in this command-clause become members of the IRD-schema-structure specified in the command and all supersets of this IRD-schema-structure.

#### Format

```
append-clause ::=
    append-word           5
    [ members-word       5
      of-word            M1.4.3
    ]
    meta-entity-access-name-list 5
```

**Syntax Rules** None

**General Rules**

- (1) Each meta-entity-access-name specified in meta-entity-access-name-list shall identify a meta-entity which exists in the IRD Schema.
- (2) Each meta-entity identified by a meta-entity-access-name in meta-entity-access-name-list shall be of type IRD-SCHEMA-STRUCTURE.

**Error and Warning Conditions**

- (1) Error E01067: Meta-entity does not exist or is not visible. See General Rule (1).
- (2) Error E04002: Specified meta-entity is not an IRD-schema-structure. See General Rule (2).

**7.2 Include Command-Clause**

**Function** To identify those meta-entities which are to become either a member or a subset of an IRD-schema-structure specified in an add-structure-command or modify-structure-command.

**Format**

```

include-clause ::=
    include-word           5
    include-option

include-option ::=
    members-clause        7.5
    | subsets-clause      7.6
    | members-and-subsets

members-and-subsets ::=
    members-clause        7.5
    subsets-clause        7.6
    |
    subsets-clause        7.6
    members-clause        7.5
    
```

**Syntax Rules** None

**General Rules** None

**Error and Warning Conditions** None

**7.3 Disconnect Command-Clause**

**Function** To specify, within a modify-structure-command, that the meta-relationship from the IRD-schema-structure specified in the command to each IRD-schema-structure specified in the subsets-clause is to be deleted.

**Format**

```

disconnect-clause ::=
    disconnect-word           5
    subsets-clause           7.6
    
```

**Syntax Rules**    None

**General Rules**

- (1) Each IRD-schema-structure specified in the subsets-clause shall be a direct subset of the IRD-schema-structure specified in the modify-structure-command.

**Error and Warning Conditions**

- (1) Error E04012: Specified IRD-schema-structure is not a direct subset of IRD-schema-structure being maintained. See General Rule (1).

**7.4 Exclude Command-Clause**

**Function**    To identify meta-entities which are to be removed from the membership or subsets of an IRD-schema-structure specified in a modify-IRD-schema-structure command.

If *S* is the IRD-schema-structure specified in a modify-IRD-schema-structure command, and *M* is a member of *S* specified in the members-clause subclause, then the meta-relationship from *S* SET-OF *M* shall be deleted.

If *S* is the IRD-schema-structure specified in the modify-IRD-schema-structure command, *S'* a direct subset of *S*, and *M'* a member of *S'*, then the following meta-relationships shall be deleted:

```

S HAS-SUBSET S'
S SET-OF M
    
```

Note that the exclude-clause has no effect on supersets of the maintained IRD-schema-structure.

**Format**

```

exclude-clause ::=
    exclude-word           5
    exclude-option
exclude-option ::=
    members-clause       7.5
    | subsets-clause     7.6
    | members-and-subsets
members-and-subsets ::=
    members-clause       7.5
    subsets-clause       7.6
    |
    
```

subsets-clause	7.6
members-clause	7.5

**Syntax Rules** None

**General Rules**

- (1) Each IRD-schema-structure specified in the subsets-clause shall be a direct subset of the IRD-schema-structure specified in the modify-IRD-schema-structure command.
- (2) Each meta-entity specified in the members-clause subclause shall be a member of the IRD-schema-structure specified in the modify-structure command.

**Error and Warning Conditions**

- (1) Error E04012: Specified IRD-schema-structure is not a direct subset of IRD-schema-structure being maintained. See General Rule (1).
- (2) Error E04013: Specified meta-entity is not a member of the IRD-schema-structure being maintained. See General Rule (2).

**7.5 Members Command-Clause**

**Function** To identify meta-entities to be included or excluded from the membership of an IRD-schema-structure.

**Format**

members-clause ::=	
members-word	5
meta-entity-access-name-list	5

**Syntax Rules** None

**General Rules**

- (1) Each meta-entity-name specified in meta-entity-access-name-list shall identify a meta-entity which exists in the IRD Schema.
- (2) Each meta-entity identified in meta-entity-access-name-list shall be of type ENTITY-TYPE or RELATIONSHIP-TYPE.

**Error and Warning Conditions**

- (1) Error E01067: Meta-entity does not exist or is not visible. See General Rule (1).
- (2) Error E04014: Specified meta-entity must be a relationship-type or entity-type. See General Rule (2).

**7.6 Subsets Command-Clause**

**Function** To identify those IRD-schema-structures which are either to become direct subsets

of a given IRD-schema-structure or to be excluded or disconnected from an IRD-schema-structure specified.

**Format**

```
subsets-clause ::=
    subsets-word           5
    meta-entity-access-name-list 5
```

**Syntax Rules**    None

**General Rules**

- (1) Each meta-entity-access-name specified in meta-entity-access-name-list shall identify a meta-entity which exists in the IRD Schema.
- (2) Each meta-entity identified in meta-entity-access-name-list shall be of type IRD-SCHEMA-STRUCTURE.

**Error and Warning Conditions**

- (1) Error E01067: Meta-entity does not exist or is not visible. See General Rule (1).
- (2) Error E04002: Specified meta-entity is not an IRD-schema-structure. See General Rule (2).

## 8 IRD Schema Structure Manipulation Panel Trees

This section describes the panel trees which shall be available for performing IRD Schema Structure Manipulation functions.

**Description of the Panel Trees**

The panels and panel trees available for performing IRD Schema Structure Manipulation functions adhere to the same conventions as those provided in Module 1 of the standard. For a full description of these Panel Interface concepts and conventions, see Module 1, Section 8.

There shall exist panel trees to support the following IRD Schema Structure Manipulation commands:

```
add-IRD-schema-structure
modify-IRD-schema-structure
delete-IRD-schema-structure
copy-IRD-schema-structure
install-IRD-structure
```

## 9 Modifications to the IRD Schema Definition

This section describes the modifications to the IRD Schema Definition.

### 9.1 New Meta-Entity-Types

One new meta-entity-type, IRD-SCHEMA-STRUCTURE, is defined as follows:

#### IRD-SCHEMA-STRUCTURE

**Definition:** A set of entity-types and relationship-types which is subject to the following constraint:

If a relationship-type is within the IRD-schema-structure, so is each of the associated entity-types.

Since IRD-schema-structures are a set, the relationship-types and entity-types which compose it are called its members.

The membership of an IRD-schema-structure is defined by meta-relationships of the following types:

IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE

IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE.

It is possible for each member of one IRD-schema-structure be a member of another IRD-schema-structure. Where this is intentional, meta-relationships of the type IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE shall be used to declare this.

Associated meta-attribute-types:

ADDED-BY

MODIFIED-BY

NUMBER-OF-TIMES-MODIFIED

PURPOSE

The meta-attribute-types, ADDED-BY, MODIFIED-BY, and NUMBER-OF-TIMES-MODIFIED shall all be singular and system-maintained.

The PURPOSE meta-attribute-type is singular and optional.

Associated meta-attribute-group-types:

DATE-TIME-ADDED

DATE-TIME-MODIFIED

Both of these shall be singular, and their components shall be system-maintained.

### 9.2 New Meta-Relationship-Types

Four new meta-relationship-types are defined:

IRD - PARTITION - GREATER - THAN - IRD - PARTITION

IRD - SCHEMA - STRUCTURE - SET - OF - ENTITY - TYPE

IRD - SCHEMA - STRUCTURE - SET - OF - RELATIONSHIP - TYPE

IRD - SCHEMA - STRUCTURE - HAS - SUBSET - IRD - SCHEMA - STRUCTURE

Their respective definitions and a description of their characteristics follow:

#### IRD-PARTITION-GREATER-THAN-IRD-PARTITION

**Definition:** Meta-relationships of this type define the Life-Cycle-Phase-Hierarchy.  
Meta-entity-type-1: IRD-PARTITION

Meta-entity-type-2: IRD-PARTITION  
Meta-relationship-class-type: GREATER-THAN

Ratio: (0,1:0,m)

Sequenced: NO.

Unique with respect to meta-entities: YES.

Associated meta-attribute-types:

SYSTEM-LOCK

IMPLEMENTATION-LOCK

Both of these meta-attribute-types shall be system-maintained.

Associated meta-attribute-group-types: None

Constraints: The following constraints apply:

- (1) A meta-relationship of this type shall be established between two meta-entities of type IRD-PARTITION only if each meta-entity has a LIFE-CYCLE-PHASE-CLASS meta-attribute.
- (2) The meta-entities in any meta-relationship of this type shall be distinct.
- (3) If the first meta-entity is the IRD-partition which has the meta-attribute LIFE-CYCLE-PHASE-CLASS equal to ARCHIVED, then there shall only be one such meta-relationship. In this meta-relationship, the second meta-entity shall have the meta-attribute LIFE-CYCLE-PHASE-CLASS equal to CONTROLLED.
- (4) A meta-relationship of this type shall not be deleted if it is the one identified in (3) above, or if the IRD-partition represented by either meta-entity has an entity in it, or if any IRD-partition subordinate to the second meta-entity has an entity in it. In these cases, the SYSTEM-LOCK meta-attribute equals ON.
- (5) The meta-relationships shall be acyclic. That is, given meta-entities  $P$  and  $P'$  of type IRD-PARTITION, if the meta-relationship  $P$  GREATER-THAN  $P'$  exists, both of the following shall be satisfied:
  - (a)  $P$  and  $P'$  are distinct.
  - (b) There is no sequence of distinct meta-entities of type IRD-PARTITION  $P_1 \dots P_n$  such that the following meta-relationships exist:

$P' \text{ GREATER-THAN } P_1$

...

$P_i \text{ GREATER-THAN } P_{i+1}$

...

$P_n \text{ GREATER-THAN } P_{n+1}$

#### IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE

Definition: A meta-relationship of this type identifies a entity-type as being a member of an IRD-schema-structure.

Meta-entity-type-1: IRD-SCHEMA-STRUCTURE

Meta-entity-type-2: ENTITY-TYPE

Meta-relationship-class-type: SET-OF

Ratio: (0,n:0,m)

Sequenced: NO.

Unique with respect to meta-entities: YES.

Associated meta-attribute-types:

SYSTEM-LOCK

IMPLEMENTATION-LOCK

Both of these meta-attribute-types shall be singular and system-maintained.

Associated meta-attribute-group-types: None

Constraints: The following constraints apply:

- (1) The completeness-constraint defined in 4.2.2.3.
- (2) The subset-membership-constraint defined in 4.2.2.3.



- (3) The SYSTEM-LOCK meta-attribute equals ON whenever both meta-entities are in the CONTROLLED IRD Schema life-cycle-phase.

**IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE**

Definition: A meta-relationship of this type identifies a relationship-type as being a member of an IRD-schema-structure.

Meta-entity-type-1: IRD-SCHEMA-STRUCTURE

Meta-entity-type-2: RELATIONSHIP-TYPE

Meta-relationship-class-type: SET-OF

Ratio: (0,n:0,m)

Sequenced: NO.

Unique with respect to meta-entities: YES.

Associated meta-attribute-types:

SYSTEM-LOCK

IMPLEMENTATION-LOCK

Both of these meta-attribute-types shall be singular and system-maintained.

Associated meta-attribute-group-types: None

Constraints: The following constraints apply:

- (1) The completeness-constraint defined in 4.2.2.3.
- (2) The subset-membership-constraint defined in 4.2.2.3.
- (3) The SYSTEM-LOCK meta-attribute equals ON whenever both meta-entities are in the CONTROLLED IRD Schema life-cycle-phase.

**IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE**

Definition: A meta-relationship of this type identifies the IRD-schema-structure identified by the second meta-entity as being a direct subset of the IRD-schema-structure identified by the first meta-entity.

Meta-entity-type-1: IRD-SCHEMA-STRUCTURE

Meta-entity-type-2: IRD-SCHEMA-STRUCTURE

Meta-relationship-class-type: HAS-SUBSET

Ratio: (0,n:0,m)

Sequenced: NO.

Unique with respect to meta-entities: YES.

Associated meta-attribute-types:

SYSTEM-LOCK

IMPLEMENTATION-LOCK

Both of these meta-attribute-types are singular and system-maintained.

Associated meta-attribute-group-types: None

Constraints: The following constraints apply:

- (1) The acyclic nesting constraint defined in 4.2.2.3.
- (2) The subset-membership constraint defined in 4.2.2.3.
- (3) The SYSTEM-LOCK meta-attribute equals ON whenever both meta-entities are in the CONTROLLED IRD Schema life-cycle-phase.

**10 Modifications to Module 1**

This section identifies and specifies all changes to the functionality of Module 1 required by this Module.

Changes to the command language specification are identified first. Changes are specified by the groups of commands in Module 1. Only those commands which require modification are identified.

For each command, required changes are identified in:

- (1) Format
- (2) Syntax Rules
- (3) General Rules
- (4) Actions

Furthermore, changes in Security Rules are also specified. These changes presume that facilities of Module 3, "IRDS Security" are in effect. If the facilities of Module 3 are not in effect, the changes to Security Rules are not applicable.

Changes are identified as being in one of the following classes:

- (1) *Additions.* The additional wording or syntax does not require any changes to any other rules and/or actions. Additional syntax will be identified by a comment identifying new syntax. Additional rules and/or actions will be identified in one of two ways:
  - (a) If only one rule or action is to be added, no special designation will be used.
  - (b) A "(+digit)" indicates that the rule is one of several rules or actions. In this case, the digit indicates the relative position of the new rule or action in the series.
- (2) *Modifications.* The command format or a given rule and/or action has been changed. The overall sense of the rule or action, however, remains the same. For command formats, the change is minor.
- (3) *Replacements.* A given rule or set of syntax-states have been substantially changed.
- (4) *Deletions.* A given command format, rule, or action is deleted in its entirety.

For each command, any changes in the corresponding panel-tree are also identified.

After the changes to the commands, changes in command-clauses, common-syntax, and other descriptive material not identified previously are presented.

### 10.1 Modifications to IRD Schema Commands

Changes shall be required in each of the following commands:

- add-meta-entity-command
- modify-meta-entity-command
- delete-meta-entity-command
- add-meta-relationship-command
- modify-meta-relationship-command
- delete-meta-relationship-command
- modify-meta-entity-life-cycle-phase-command

copy-meta-entity-command  
activate-IRD-command

In each case, the modifications are minor.

### 10.1.1 Modifications to Add Meta-Entity Command

#### General Rules

(+1) The type of the specified meta-entity shall not be IRD-SCHEMA-STRUCTURE.

#### Error Conditions

(+1) Error E04015: Command not valid for meta-entities of type IRD-SCHEMA-STRUCTURE.

### 10.1.2 Modifications to Modify Meta-Entity Command

#### General Rules

(+1) The type of the specified meta-entity shall not be IRD-SCHEMA-STRUCTURE.

#### Error Conditions

(+1) Error E04015: Command not valid for meta-entities of type IRD-SCHEMA-STRUCTURE.

### 10.1.3 Modifications to Delete Meta-Entity Command

#### General Rules

(+1) The type of the specified meta-entity shall not be IRD-SCHEMA-STRUCTURE.

#### Error Conditions

(+1) Error E04015: Command not valid for meta-entities of type IRD-SCHEMA-STRUCTURE.

### 10.1.4 Modifications to Add Meta-Relationship Command

#### General Rules

(+1) The specified meta-relationship shall not be of one of the following types:

IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE  
IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE  
IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE

#### Error Conditions

(+1) Error E04016: Command not valid for meta-relationships of specified type.

### 10.1.5 Modifications to Modify Meta-Relationship Command

#### General Rules

(+1) The specified meta-relationship shall not be of one of the following types:

IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE  
IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE  
IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE

### **Error Conditions**

- (+1) Error E04016: Command not valid for meta-relationships of specified type.

### **10.1.6 Modifications to Delete Meta-Relationship Command**

#### **General Rules**

- (+1) The specified meta-relationship shall not be of one of the following types:

IRD-SCHEMA-STRUCTURE-SET-OF-ENTITY-TYPE  
IRD-SCHEMA-STRUCTURE-SET-OF-RELATIONSHIP-TYPE  
IRD-SCHEMA-STRUCTURE-HAS-SUBSET-IRD-SCHEMA-STRUCTURE

- (+2) A meta-relationship of type RELATIONSHIP-TYPE-CONNECTS-ENTITY-TYPE cannot be deleted if the specified relationship-type meta-entity is a member of an IRD-schema-structure.

### **Error Conditions**

- (+1) Error E04016: Command not valid for meta-relationships of specified type. See General Rule (+1).
- (+2) Error E04017: Relationship-type shall be excluded from IRD-schema-structure(s) prior to deleting specified meta-relationship. See General Rule (+2).

### **10.1.7 Modifications to Modify Meta-Entity Life-Cycle-Phase Command**

#### **Modification of General Rule (8)**

- (8) Each meta-entity specified in meta-entity-name-list shall be one of the following types:

IRD-SCHEMA-STRUCTURE /\* new \*/  
RELATIONSHIP-TYPE  
RELATIONSHIP-CLASS-TYPE  
ENTITY-TYPE  
VARIATION-NAMES-DATA  
ATTRIBUTE-GROUP-TYPE  
ATTRIBUTE-TYPE  
ATTRIBUTE-TYPE-VALIDATION-DATA

### **10.1.8 Modifications to Copy Meta-Entity Command**

#### **General Rules**

- (+1) The type of the specified meta-entity shall not be IRD-SCHEMA-STRUCTURE.

### **Error Conditions**

- (+1) Error E04015: Command not valid for meta-entities of type IRD-SCHEMA-STRUCTURE.

### 10.1.9 Modifications to Activate IRD Command

#### General Rules

(+1) In the CONTROLLED IRD Schema life-cycle-phase, there shall be two meta-entities of type IRD-SCHEMA-STRUCTURE.

(a) One meta-entity shall have the assigned-access-name IRD-STRUCTURE.

(b) The other shall have the assigned-access-name PHASE-SENSITIVITY-STRUCTURE.

These meta-entities shall be called the IRD Structure and the Phase Sensitivity Structure, respectively.

(+2) The Phase Sensitivity Structure shall be a (direct or indirect) subset of the IRD Structure.

#### Error Conditions

(+1) Error E04018: IRD Structure not defined. See General Rule (+1)(a).

(+2) Error E04019: Phase Sensitivity Structure not defined. See General Rule (+1)(b).

(+3) Error E04020 Phase Sensitivity Structure not subset of IRD Structure. See General Rule (+2).

### 10.2 Modifications to the IRD Commands

Modifications shall be required in each of the following commands:

add-entity-command  
 modify-entity-command  
 delete-entity-command  
 add-relationship-command  
 modify-relationship-command  
 delete-relationship-command  
 copy-entity-command  
 output-impact-of-change-command

#### 10.2.1 Modifications to Add Entity Command

##### Replacement for Syntax Rule (2)

(2) The rules for specifying a version identifier with new-entity-access-name shall be as follows:

(a) If entity-type-clause identifies an entity-type which is a member of the Phase Sensitivity Structure, then the version identifier shall be optional. Otherwise, version identifier shall not be allowed.

(b) If specified, the version-identifier shall specify a valid variation-name for the entity-type.

(c) If specified, the version-identifier shall not specify a revision-number.

### General Rules

- (+1) An entity shall be created only in an uncontrolled IRD life-cycle-phase or an IRD-partition which is not an IRD life-cycle-phase.

### Security Rules

- (+1) If the entity-type of the entity being added is a member of the Phase Sensitivity Structure, the effective IRD view shall be associated with an uncontrolled IRD life-cycle-phase.
- (+2) If the entity-type of the entity being added is not a member of the Phase Sensitivity Structure, the effective IRD view shall be associated with the IRD-partition reserved for entities of that type.

### Error Conditions

- (+1) Error E01005: Invalid Name Format. See Syntax Rule (2)(a). A version-identifier was specified for an entity whose type is not in the Phase Sensitivity Structure.
- (+2) Error E01006: Invalid version-identifier. See Syntax Rules (2)(b) and (2)(c).
- (+3) Error E01115: Effective IRD-view not associated with correct IRD partition. See General Rule (+1) and Security Rules (+1) and (+2).

## 10.2.2 Modifications to Modify Entity Command

### General Rules

- (+1) An entity whose type is a member of the Phase Sensitivity Structure shall be created or modified only in an uncontrolled IRD life-cycle-phase.
  - (a) If existing-entity-name identifies an entity in the controlled or archived IRD life-cycle-phase, the new-version-clause shall be required.
  - (b) If the entity is not in an IRD life-cycle-phase, new-version-clause shall not be allowed.
- (+2) Creation of relationships as required by the new-version-clause shall not violate IRD life-cycle-phase integrity. It therefore follows that if the entity identified by existing-entity-access-name depends on entities in the archived IRD life-cycle-phase, corresponding relationships cannot be produced. In this case, a warning message shall be produced.

### Security Rules

- (+1) If the type of the entity being maintained is a member of the Phase Sensitivity Structure, the effective IRD-view shall be associated with an uncontrolled IRD life-cycle-phase. Otherwise, the effective IRD-view shall be associated with the IRD-partition which contains the entity being modified.

### Actions

The last paragraph of Action (2) now reads:

For each relationship which has the specified entity only as the second entity, a new relationship of the same type shall be created provided that the corresponding relationship-type is neither sequenced nor phase-related. A sequenced relationship-type is defined by a meta-entity of type RELATIONSHIP-TYPE with the SEQUENCED meta-attribute equal to YES. (A phase-related relationship-type is defined by a meta-entity of type RELATIONSHIP-TYPE, which is a member of the Phase Sensitivity Structure.) The first entity and the attributes of the corresponding relationships shall be identical.

### **Error Conditions**

- (+1) Error E01022: Multiple versions not allowed for entities of entity-type. See General Rule (1).
- (+2) Error E01115: Effective IRD-view not associated with correct IRD partition. See General Rule (+1) and Security Rule (+1).
- (+3) Warning W04002: Corresponding relationship would violate IRD life-cycle-phase integrity and cannot be created. See General Rule (2).

### **10.2.3 Modifications to Delete Entity Command**

#### **General Rules**

- (+1) If with-relationships-clause is specified, then each entity to be deleted cannot have any other entities dependent on it. (An entity *A* depends on entity *B* if there exists a forward phase-related relationship *A R B* where *R* is the corresponding type.)

#### **Security Rules**

- (+1) If with-relationships-clause is specified, then the effective IRDS user shall have authorization to delete the following relationships:
  - (a) Every phase-related relationship in which the entity to be deleted is the first entity of the relationship.
  - (b) Every other relationship which is not phase-related in which the entity to be deleted participates.

#### **Replacement for Action (1)**

- (1) The command shall process according to the rules stated in the General Rules, Section 5 of Module 1. For each entity specified or selected in selection option:
  - (a) Failure to comply with any of the above rules shall cause the deletion of that entity to fail. Processing shall continue with the next specified or selected entity.
  - (b) If no errors were encountered for the entity to be deleted:
    - (i) If with-relationships was specified, then the entity and all the relationships in which it participates shall be deleted.
    - (ii) If with-relationships was not specified, the entity shall be deleted. (In this case, there are no relationships to be deleted.)

### **Error Conditions**

- (+1) Error E04021: Specified entity cannot be deleted. Other entities depend on it. See General Rule (+1).

### **10.2.4 Modifications to Add Relationship Command**

#### **General Rules**

- (+1) If the relationship-type is phase-related, the first entity of the specified relationship shall be in an uncontrolled IRD life-cycle-phase.
- (+2) The IRD life-cycle-phase integrity constraint shall not be violated. That is:

If the specified relationship is *E1 R E2*, where:

*E1* is the first entity,  
*E2* is the second entity,  
*R* is the corresponding relationship-type,

and *R* is phase-related, then

*E2* shall be in either an uncontrolled or the controlled IRD life-cycle-phase. (*E1* shall be in an uncontrolled IRD life-cycle-phase by the previous rule.)

#### **Security Rules**

- (+1) If the specified relationship is phase-related, the effective IRD-view shall have an IRD-PARTITION-NAME equal to the assigned-access-name of an IRD-partition which identifies an uncontrolled IRD life-cycle-phase.
- (+2) If the relationship to be added is of a phase-related relationship-type, then the effective IRD-view shall authorize MODIFY-PERMISSION for entities of the first entity-type in the relationship-type.

### **Error Conditions**

- (+1) Error E04022: Cannot add a phase-related relationship which affects a controlled or archived entity. See General Rule (+1) and Security Rule (+1).
- (+2) Error E04023: Specified relationship would violate IRD life-cycle-phase integrity. See General Rule (+2).
- (+3) Error E01115: Effective IRD-view not associated with correct IRD partition. See Security Rule (+1).
- (+4) Error E03011: User not authorized to perform specified actions by the effective IRD-view. See Security Rule (+2).

### **10.2.5 Modifications to Modify Relationship Command**

#### **Security Rules**

- (+1) If the specified relationship is phase-related, the effective IRD-view shall have an IRD-



PARTITION-NAME equal to the assigned-access-name of an IRD-partition which identifies an uncontrolled IRD life-cycle-phase.

- (+2) If the specified relationship is phase-related, then the effective IRD-view shall authorize MODIFY-PERMISSION for entities of the first entity-type in the relationship-type.

#### **Error Conditions**

- (+1) Error E01115: Effective IRD-view not associated with correct IRD partition. See Security Rule (+1).
- (+2) Error E03011: User not authorized to perform specified actions by the effective IRD-view. See Security Rule (+2).

### **10.2.6 Modifications to Delete Relationship Command**

#### **Security Rules**

- (+1) If
  - (a) effective IRD-view has an IRD-PARTITION-NAME equal to the assigned-access-name of an IRD-partition which identifies an uncontrolled IRD life-cycle-phase,
  - and
  - (b) the specified relationship is phase-related
 then the effective IRD-view shall authorize MODIFY-PERMISSION for entities of the first entity-type in the relationship-type.
- (+2) If
  - (a) effective IRD-view has an IRD-PARTITION-NAME equal to the assigned-access-name of an IRD-partition which identifies either the controlled or archived IRD life-cycle-phase,
  - and
  - (b) the specified relationship is phase-related
 then the effective IRD-view shall authorize DELETE-PERMISSION for entities of the first entity-type in the relationship-type.

#### **Error Conditions**

- (+1) Error E03011: User not authorized to perform specified actions by the effective IRD-view. See Security Rules (+1) and (+2).

### **10.2.7 Modifications to Modify Entity Life-Cycle-Phase Command**

#### **Syntax Rules**

- (+1) The following defines the allowed transitions of entities across IRD life-cycle-phases.

CURRENT

any uncontrolled phase  
any uncontrolled phase  
the controlled phase  
the archived phase  
the controlled phase

NEW

any other uncontrolled phase  
the controlled phase  
the archived phase  
the controlled phase  
any uncontrolled phase

**General Rules**

- (+1) IRD life-cycle-phase integrity shall be preserved. This means that if an entity is moved to the controlled IRD life-cycle-phase, each entity which it depends on shall already be in the controlled IRD life-cycle-phase. The allowed phase-transitions identified in the Syntax Rules also support this rule.
- (+2) This command shall be valid only for entities whose corresponding type is a member of the Phase Sensitivity Structure.

**Error Conditions**

- (+1) Error E04024: Invalid phase transition. See Syntax Rule (+1).
- (+2) Error E04025: Entity cannot be moved since it depends on another entity which is not in appropriate IRD life-cycle-phase. See General Rule (+1).
- (+3) Error E04026: Specified entity is of type for which command is not valid.

**10.2.8 Modifications to Copy Entity Command**

**General Rules**

- (+1) It shall not be possible to use this command to create an entity in the controlled or archived IRD life-cycle-phase.
- (+2) If with-relationships-clause is specified, IRD life-cycle-phase integrity shall be preserved. It therefore follows that if the entity identified by existing-entity-access-name depends on entities in the archived IRD life-cycle-phase, corresponding relationships cannot be produced. In this case, a warning message shall be produced.
- (+3) New-version-clause shall be allowed only for entities whose type is a member of the Phase Sensitivity Structure.

**Security Rules**

- (+1) The effective IRD-view shall not have the IRD-PARTITION-NAME attribute equal to the assigned-access-name of the meta-entity which represents either the controlled or archived IRD life-cycle-phase.

**Replacement for Action (3)**

For each relationship which has the existing entity only as the second entity, a new relationship of the same type shall be created with the new entity as the second entity, provided that the corresponding relationship-type is neither sequenced nor phase-related. (A sequenced relationship-type is defined by a meta-entity of type RELATIONSHIP-TYPE

with the SEQUENCED meta-attribute equal to YES. A phase-related relationship-type is defined by a meta-entity of type RELATIONSHIP-TYPE, which is a member of the Phase Sensitivity Structure.) The first entity and the attributes of the corresponding relationships will not change.

### Error Conditions

- (+1) Error E01115: Effective IRD-view not associated with correct IRD-partition. See General Rule (+1) and Security Rule (+1).
- (+2) Warning W04002: Corresponding relationship would violate IRD life-cycle-phase integrity and cannot be created. See General Rule (+2).
- (+3) Error E01022: Multiple versions not allowed for entities of entity-type. See General Rule (+3).

### 10.2.9 Output Impact-of-Change Command

#### Replacement for General Rule (1)

- (1) In the following definition, the notation  $E1 R E2$  shall denote the forward relationship of type  $R$  in which  $E1$  is the first entity and  $E2$  is the second entity.

For any given entity  $E$ , the set of entities potentially impacted by a change in  $E$  is defined to be the set of all entities  $E'$  such that one of the following is satisfied:

- (a) Either  $E R E'$  or  $E' R E$  for some relationship-type  $R$  which is not phase-related.
- (b)  $E'$  depends on  $E$  (as specified in the definition of Life-Cycle-Phase Integrity.)

### 10.3 Modifications to IRD-IRD Interface Commands

Changes shall be required in each of the following commands:

```
create-IRD command
check-IRD-schema-compatibility command
```

#### 10.3.1 Modifications to Create IRD Command

##### Actions

- (+1) When the IRDS has the facilities of this Module in effect, the following IRD Schema descriptors shall also be created in the IRD Schema:
  - (a) The following meta-entities shall be created in the CONTROLLED IRD Schema life-cycle-phase:

```
IRD-STRUCTURE(1)
PHASE-SENSITIVITY-STRUCTURE(1)
```

- (b) The following meta-relationship shall also be created:

```
IRD-STRUCTURE(1) HAS-SUBSET PHASE-SENSITIVITY-STRUCTURE(1)
```

### 10.3.2 Modifications to Check IRD Schema Compatibility Command

#### Replacement for General Rule (4)

- (4) IRD Schema compatibility shall be determined by comparing the following sets of meta-entities in the source and target IRD Schema:
- (a) Those meta-entities of type ENTITY-TYPE and RELATIONSHIP-TYPE which are members of the IRD Structure.
  - (b) Each meta-entity of one of the following types which are either directly or indirectly associated with the meta-entities in (i) above:

RELATIONSHIP-CLASS-TYPE  
VARIATION-NAMES-DATA  
ATTRIBUTE-GROUP-TYPE  
ATTRIBUTE-TYPE  
ATTRIBUTE-TYPE-VALIDATION-DATA  
ATTRIBUTE-TYPE-VALIDATION-PROCEDURE

- (c) Meta-entities of the following types:

QUALITY-INDICATOR  
IRDS-DEFAULTS

The set of meta-entities as defined above is called the IRD Schema comparison set. The sets for the source and target IRD Schemas are called the source-IRD Schema comparison set and target-IRD Schema comparison set respectively. Two meta-entities, one in each of the above sets, are said to be comparable if they have the same assigned-access-name. Each meta-entity in the source-IRD Schema comparison set shall have a comparable meta-entity in the target-IRD Schema comparison set.

### 10.4 Modifications in Command-Clauses for Module 1 Commands

Modifications to the following existing command-clauses are required:

with-relationships-clause  
life-cycle-phase-restriction-clause

#### 10.4.1 Modifications to With Relationships Command-Clause

##### Replacement for Function

For a copy-entity command, to specify that new relationships shall be created when the new entity is created.

For a delete-entity-command, to specify that all relationships shall be deleted when the entity is deleted.

#### 10.4.2 Modifications to Life-Cycle-Phase Restriction

This command-clause now supports all relational-operators. Previously, it was restricted to equals-operator and not-equal-operator.

**Replacement for Format**

```

life-cycle-phase-restriction-clause ::=
    life-cycle-phase-word      /* unchanged */      M1.4.3
    relational-operator        /* changed */         M1.4.3
    life-cycle-phase-name     /* unchanged */
life-cycle-phase-name ::=
    meta-entity-assigned-access-name                M1.4.4

```

**Modification of Syntax Rule (1)**

- (1) Life-cycle-phase-name shall be the name of a meta-entity of type IRD-PARTITION which is in the Life Cycle Phase Hierarchy.

**Replacement for General Rule (1)**

- (1) The relational operators ">" and "<" shall be interpreted as superior-to and inferior-to in the Life-Cycle-Phase-Hierarchy. The relational-operators "=" and "<>" retain their conventional meanings of equality and inequality respectively.

**10.5 Modifications to the Minimal IRD Schema**

The meta-entities

```

IRD-STRUCTURE(1)
PHASE-SENSITIVITY-STRUCTURE(1)

```

shall be created in the CONTROLLED IRD Schema life-cycle-phase.

Both IRD-STRUCTURE(1) and PHASE-SENSITIVITY-STRUCTURE(1) shall be of type IRD-SCHEMA-STRUCTURE. Each shall have the SYSTEM-LOCK meta-attribute equal to ON. Both may have implementation-defined PURPOSE meta-attributes.

If only this Module is installed with Module 1, when the IRD is activated:

- (1) IRD-STRUCTURE(1) shall have the following members:

The entity-types:

```

IRDS-USER
IRD-VIEW
IRD-SCHEMA-VIEW

```

The relationship-types:

```

IRD-USER-HAS-IRD-VIEW
IRD-USER-HAS-IRD-SCHEMA-VIEW

```

- (2) PHASE-SENSITIVITY-STRUCTURE(1) shall have no members.

If in addition to Modules 1 and 4, Module 2 is also installed, when the IRD is activated:

- (1) The entity-types and the relationship-types specified in Section 5.1 and 5.2 of Module 2, respectively, shall also be members of IRD-STRUCTURE(1), and
- (2) The entity-types specified in Section 5.1 of Module 2 shall be members of PHASE-SENSITIVITY-STRUCTURE(1).

If in addition to Modules 1 and 4, Module 3 is also installed, when the IRD is activated:

- (1) The entity-type ACCESS-CONTROLLER shall be a member of IRD-STRUCTURE(1).
- (2) PHASE-SENSITIVITY-STRUCTURE(1) shall have no additional member.

The meta-entity, STANDARD-RESERVED-NAMES shall be modified to have components IRD-STRUCTURE and PHASE-SENSITIVITY-STRUCTURE of the repeating meta-attribute-type IRDS-RESERVED-NAMES.

There shall be three additional meta-relationships:

```
IRD-STRUCTURE(1) HAS-SUBSET PHASE-SENSITIVITY-STRUCTURE(1)
archived-phase GREATER-THAN controlled-phase
controlled-phase GREATER-THAN uncontrolled-phase
```

Here the labels "archived-phase", "controlled-phase", and "uncontrolled-phase" signify the implementation-defined assigned-access-names for the meta-entities of type IRD-PARTITION, automatically created by the IRDS when a create-IRD command is executed. Note that for these IRD-partitions:

- (1) Uncontrolled-phase has the LIFE-CYCLE-PHASE-CLASS meta-attribute equal to UNCONTROLLED.
- (2) Controlled-phase has the LIFE-CYCLE-PHASE-CLASS meta-attribute equal to CONTROLLED.
- (3) Archived-phase has the LIFE-CYCLE-PHASE-CLASS meta-attribute equal to ARCHIVED.

## 10.6 Conversion Requirements

The implementation shall provide a mechanism for an installation to install the functionality of this Module on an existing IRD.

One problem which shall be addressed relates to having entities in different IRD life-cycle-phase. Under the rules of this Module, each IRD life-cycle-phase which contains at least one entity shall be part of the Life-Cycle-Phase Hierarchy. This involves having meta-relationships between the meta-entities which represent IRD life-cycle-phases. Thus the implementation shall provide a mechanism for these meta-relationships to be established between existing IRD life-cycle-phases prior to Module 4 becoming fully functional. The responsibility for the solution of this problem belongs to the implementor of this Module.

## 11 Modifications to Module 2

Module 2 of the standard IRDS, "The Basic Functional IRD Schema", is independent of this Module of the standard IRDS. There are no modifications to Module 2 required by this Module of the standard IRDS.

## 12 Modifications to Module 3

Module 3 of the standard IRDS, "IRDS Security", is independent of this Module of the standard IRDS. There are no modifications to Module 3 required by this Module of the standard IRDS.

## 13 Cross-Reference Indices

Three cross-reference indices are provided:

- (1) A command index.
- (2) A command-clause index.
- (3) An error and warning condition index.

The command and command-clause indices provide an alphabetical cross reference of commands and command-clauses respectively which are defined. The error and warning condition usage index identifies where each error and warning is used in this Module. The scope of each index is this Module only.

### 13.1 Alphabetic Index of Commands

<u>Command</u>	<u>Subsection</u>
add-IRD-schema-structure-command	6.1
copy-IRD-schema-structure-command	6.4
delete-IRD-schema-structure-command	6.3
install-IRD-structure-command	6.5
modify-IRD-schema-structure-command	6.2

### 13.2 Alphabetic Index of Command-Clauses

<u>Command-Clause</u>	<u>Subsection-Number</u>
append-command-clause	7.1
disconnect-command-clause	7.3
exclude-command-clause	7.4
include-command-clause	7.2
members-command-clause	7.5
subsets-command-clause	7.6

**13.3 Error and Warning Condition Usage Index**

<u>Cond</u>	<u>Text / Subsection</u>
E01005	Invalid name format. 10.2.1
E01006	Invalid version identifier. 10.2.1
E01022	Multiple versions not allowed for entities of entity-type. 10.2.2    10.2.8
E01063	Effective IRD-schema-view not associated with correct IRD Schema life-cycle. 6.            6.4
E01067	Meta-entity does not exist or is not visible. 6.            7.1            7.5            7.6
E01073	Meta-entity is locked and cannot be deleted. 6.3
E01075	Cannot modify specified meta-entity. New version is required. 6.2
E01076	Meta-entity already has a descriptive-name. 6.2            6.4
E01079	Cannot delete specified meta-entity. Other meta-entities depend on specified meta-entity. 6.3
E01087	Creation of meta-relationship would violate IRD Schema life-cycle-phase integrity. 6.1            6.2            6.4
E01098	IRD must be deactivated. 6.5
E01099	Another user deactivated the IRD. 6.5
E01115	Effective IRD-view not associated with correct IRD partition. 10.2.2    10.2.4    10.2.5    10.2.8
E03002	Effective IRD-schema-view does not authorize user to use command. 6.1            6.2            6.3            6.4            6.5
E03004	User not authorized to perform specified action(s). 6.1            6.2            6.3            6.4            6.5
E03011	User not authorized to perform specified actions by the effective IRD-view. 10.2.4    10.2.5    10.2.6



- E04001 Participating entity-type for a relationship-type not specified.  
6.1 6.2
- E04002 Specified meta-entity is not an IRD-schema-structure.  
6.2 6.3 6.4 6.5 7.1 7.6
- E04003 IRD-schema-structure in command-clause is not a direct subset of IRD-schema-structure being maintained.  
6.2
- E04004 Meta-entity specified in exclude-clause is not a member of IRD-schema-structure being maintained.  
6.2
- E04005 Specified member is in a subset of maintained IRD-schema-structure and cannot be excluded.  
6.2
- E04006 Specified entity-type participates in a relationship-type which is a member of maintained IRD-schema-structure.  
6.2
- E04007 Command would create a cycle of IRD-schema-structures.  
6.2
- E04008 Assigned-access-name must be IRD-STRUCTURE.  
6.5
- E04009 Specified meta-entity not in UNCONTROLLED phase.  
6.5
- E04010 IRD-schema-structure has two distinct components with the same assigned-access-name.  
6.5
- E04012 Specified IRD-schema-structure is not a direct subset of IRD-schema-structure being maintained.  
7.3 7.4
- E04013 Specified meta-entity is not a member of the IRD-schema-structure being maintained.  
7.4
- E04014 Specified meta-entity must be a relationship-type or entity-type.  
7.5
- E04015 Command not valid for meta-entities of type IRD-SCHEMA-STRUCTURE.  
10.1.1 10.1.2 10.1.3 10.1.8
- E04016 Command not valid for meta-relationships of specified type.  
10.1.4 10.1.5 10.1.6
- E04017 Relationship-type must be excluded from IRD-schema-structure(s) prior to deleting specified meta-relationship.  
10.1.6

- E04018 IRD Structure not defined.  
10.1.9
  
- E04019 Phase Sensitivity Structure not defined.  
6.5 10.1.9
  
- E04020 Phase Sensitivity Structure not subset of IRD Structure.  
10.1.9
  
- E04021 Specified entity cannot be deleted. Other entities depend on it.  
10.2.3
  
- E04022 Cannot add a phase-related relationship which affects a controlled or archived entity.  
10.2.4
  
- E04023 Specified relationship would violate IRD life-cycle-phase integrity.  
10.2.4
  
- E04024 Invalid phase transition.  
10.2.7
  
- E04025 Entity cannot be moved. It depends on another entity which is not in appropriate  
IRD life-cycle-phase.  
10.2.7
  
- E04026 Specified entity is of type for which command is not valid.  
10.2.7
  
- W04001 Command calls for appending a superset of specified IRD-schema-structure.  
6.2
  
- W04002 Corresponding relationship would violate IRD life-cycle-phase integrity and cannot be  
created.  
10.2.2 10.2.8

# Appendixes

(These Appendixes are not part of American National Standard X3.138-1988, but are included for information only.)

## Appendix A

### Collected Syntax

```
add-IRD-schema-structure-command ::=
    add-word
    IRD-schema-structure-word
    new-meta-entity-access-name
    [ meta-entity-descriptive-name-clause ]
    [ new-meta-attributes-clause ]
    [ append-clause ]
    [ include-clause ]
    ;

append-clause ::=
    append-word
    [ members-word
      of-word
    ]
    meta-entity-access-name-list

append-word ::= APPEND

copy-IRD-schema-structure-command ::=
    copy-word
    IRD-schema-structure-word
    existing-meta-entity-access-name
    to-word
    new-IRD-schema-structure
    [ meta-entity-descriptive-name-clause ]
    ;

delete-IRD-schema-structure-command ::=
    delete-word
    IRD-schema-structure-word
    existing-meta-entity-access-name
    ;

disconnect-clause ::=
    disconnect-word
    subsets-clause

disconnect-word ::= DISCONNECT

exclude-clause ::=
    exclude-word
    exclude-option

exclude-option ::=
    members-clause
    | subsets-clause
    | members-and-subsets

include-clause ::=
    include-word
    include-option

include-option ::=
    members-clause
    | subsets-clause
    | members-and-subsets

include-word ::= INCLUDE

install-IRD-structure-command ::=
    install-word
    existing-meta-entity-access-name
    ;

install-word ::= INSTALL

IRD-schema-structure-command ::=
    add-IRD-schema-structure-command
    | modify-IRD-schema-structure-command
    | delete-IRD-schema-structure-command
    | copy-IRD-schema-structure-command
    | install-IRD-structure-command
    ;

IRD-schema-structure-word ::= IRD-SCHEMA-STRUCTURE

life-cycle-phase-name ::=
    meta-entity-assigned-access-name

life-cycle-phase-restriction-clause ::=
    life-cycle-phase-word
    relational-operator
    life-cycle-phase-name
```

```

members-and-subsets ::=
    members-clause
    subsets-clause
    |
    subsets-clause
    members-clause

members-clause ::=
    members-word
    meta-entity-access-name-list

members-word ::= MEMBERS | MEMBER

meta-entity-access-name-list ::=
    meta-entity-access-name
    [ , meta-entity-access-name-list ]

modify-IRD-schema-structure-command ::=
    modify-word
    IRD-schema-structure-word
    existing-meta-entity-access-name
    [ new-meta-entity-version-clause ]
    [ meta-entity-descriptive-name-clause ]
    [ modified-meta-attributes-clause ]
    [ disconnect-clause ]
    [ exclude-clause ]
    [ append-clause ]
    [ include-clause ]
    ;

new-IRD-schema-structure ::=
    new-meta-entity-access-name
    | new-meta-entity-version-clause

subsets-clause ::=
    subsets-word
    meta-entity-access-name-list

subsets-word ::= SUBSETS | SUBSET

```

## Appendix B

### Examples of Schema-Structure Manipulation

This Appendix gives a series of examples which demonstrate the functionality of the IRD-schema-structure manipulation commands defined in Module 4 of the standard IRDS.

Each example presented is illustrated by a figure. This figure shows:

- (1) A Venn diagram which illustrates the conditions prior to command execution;
- (2) The syntax of the command being executed;
- (3) A Venn diagram which illustrates the conditions after command execution,

The conventions used in the Venn diagrams are as follows:

- (1) An irregular octagon illustrates an IRD-schema-structure.
- (2) Members of IRD-schema-structures are indicated as follows:
  - (a) A square indicates an entity-type.
  - (b) A diamond indicates a relationship-type.

- (c) The line between a diamond and a square identifies an association between an entity-type and a relationship-type.
  - (d) The arrowhead on a line between an entity-type and a relationship-type indicates the (forward) direction of the relationship-type. The arrowhead always points to the second entity-type within the relationship-type.
- (3) IRD-schema-structure subsets are shown by nesting the octagons.

All meta-entities in the following examples are given names consisting of a single letter and a number. The letter "S" indicates an IRD-schema-structure, "E" indicates an entity-type, and "R" indicates a relationship type. For simplicity, it is assumed that all entity-types and relationship-types used in the following examples already exist in the IRD Schema.

### Example 1

Prior to the execution of the command discussed here, no IRD-schema-structure exists in the IRD Schema.

Suppose  $R1$  is a relationship-type with  $E1$  and  $E2$  as the first and second entity-types respectively. The command:

```
ADD IRD-SCHEMA-STRUCTURE S1
INCLUDE MEMBERS E1, E2, R1;
```

generates the following meta-relationships:

```
S1 SET-OF E1
S1 SET-OF E2
S1 SET-OF R1
```

Figure B1 shows a Venn diagram for  $S1$ . Note that the arrows connecting  $E1$ ,  $R1$ , and  $E2$  show the directionality of  $R1$ .

### Example 2

Assume that the IRD-schema-structure  $S1$  exists in the IRD Schema.

Suppose  $R2$  is associated with  $E3$  and  $E4$ , and  $R3$  is associated with  $E3$  and  $E5$ . Then the command:

```
ADD IRD-SCHEMA-STRUCTURE S2
INCLUDE MEMBERS E3, E4, E5, E6, R2, R3;
```

generates the following meta-relationships:

```
S2 SET-OF E3
S2 SET-OF E4
S2 SET-OF E5
S2 SET-OF E6
S2 SET-OF R2
S2 SET-OF R3
```

## APPENDIX

Figure B2 shows the Venn diagram for *S2*. Note that *E6* is not associated with any relationship-type in *S2*.

### Example 3

In both of the previous examples, the members of a new IRD-schema-structure were specified using the INCLUDE MEMBERS option. In both cases, the resulting IRD-schema-structures had no subsets.

Assume that the IRD-schema-structures *S1* and *S2* exist in the IRD Schema.

Now suppose a third IRD-schema-structure is defined as follows:

```
ADD IRD-SCHEMA-STRUCTURE S3
APPEND MEMBERS OF S1
INCLUDE SUBSETS S2;
```

The APPEND command-clause establishes meta-relationships from *S3* to each member of *S1*. That is:

```
S3 SET-OF E1
S3 SET-OF E2
S3 SET-OF R1
```

The INCLUDE SUBSETS command-clause makes *S2* a direct subset of *S1*. Thus, SET-OF meta-relationships are established from *S3* to each member of *S2*:

```
S3 SET-OF E3      S3 SET-OF E4      S3 SET-OF R2
S3 SET-OF E5      S3 SET-OF E6      S3 SET-OF R3
```

Additionally, the meta-relationship

```
S3 HAS-SUBSET S2
```

is created.

Figure B3 shows *S3* as produced by this command.

### Example 4

Note that in the previous example, although *S3* includes every member of *S1*, the meta-relationship *S3 HAS-SUBSET S1* does not exist. In this case, *S1* is called a coincidental subset of *S3*.

A user might use the APPEND command-clause as a convenience. In these cases, the user might subsequently remove some unwanted members in a modify-IRD-schema-structure command.

A coincidental subset can always be made explicit by executing a modify-IRD-schema-structure command with an INCLUDE SUBSETS command-clause.

Assume that the IRD Schema contains the IRD-schema-structures as at completion of the command of the previous example. Consider then the command:

```

MODIFY IRD-SCHEMA-STRUCTURE S3
INCLUDE
  SUBSETS S1
  MEMBERS R4, E7, E8, R5, R6, R7;

```

Note that both *SUBSETS* and *MEMBERS* can be specified in the *INCLUDE* command-clause.

Since each member of *S1* already is a member of *S3*, "INCLUDE SUBSET *S1*" just creates a single meta-relationship:

```
S3 HAS-SUBSET S1
```

The enumeration of meta-entities in the *MEMBERS* command-clause establishes the corresponding *SET-OF* meta-relationships:

```

S3 SET-OF E7
S3 SET-OF E8
S3 SET-OF R4
S3 SET-OF R5
S3 SET-OF R6
S3 SET-OF R7

```

Note that, as shown in Figure B4, the relationship-types *R4*, *R5*, and *R7* all are associated with an existing member of *S3*. Note also that the relationship-types specified here are not members of any subset of *S3*.

### Example 5

Suppose *S1*, *S2*, and *S3* are as previously shown in Figure B4.

The command:

```

MODIFY IRD-SCHEMA-STRUCTURE S2
EXCLUDE MEMBERS E5;

```

will fail, because *E5* participates in a relationship-type (*R3*) which is also a member of *S2*.

However, the command:

```

MODIFY IRD-SCHEMA-STRUCTURE S2
EXCLUDE MEMBERS E6;

```

will succeed, because *E6* does not participate in any relationship-types in *S2*. Figure B5 shows the result of this command. Note that *E6* is still a member of *S3*.

### Example 6

Just as Example 5 showed that a member can "pop up" in a nesting of IRD-schema-structures, members can also be "pulled down". Suppose *S1*, *S2*, and *S3* are as of the execution of the previous command and consider the command:

APPENDIX

```
MODIFY IRD-SCHEMA-STRUCTURE S2
INCLUDE MEMBERS E8, R7;
```

As Figure B6 shows, the membership of *S3* is unchanged by this action.

**Example 7**

The results of Example 6 did have one important consequence. Although the membership of *S3* is unchanged, the members which are excluded in a single command can change.

The command:

```
MODIFY IRD-SCHEMA-STRUCTURE S2
EXCLUDE MEMBERS R7;
```

would succeed if *S2* and *S3* were as shown in Figure B4. However, the command would fail against *S2* and *S3* as shown in Figure B5. If this were allowed, then *S2* would no longer be a subset of *S3*.

In order to eliminate *R7* from *S3* without affecting *S2* the following command are issued:

```
MODIFY IRD-SCHEMA-STRUCTURE S3
DISCONNECT S2
EXCLUDE MEMBERS R7;
```

Figure B7 shows the result of this command.

**Example 8**

Now suppose it is desired that *R7* be reinstated in *S3*. This can be done in one of two ways:

```
MODIFY IRD-SCHEMA-STRUCTURE S3
INCLUDE MEMBERS R7;
```

- or -

```
MODIFY IRD-SCHEMA-STRUCTURE S3
INCLUDE SUBSETS S2;
```

With the former command *S2* is a coincidental subset of *S3*. In the latter it again becomes an explicit subset of *S3*.

Suppose that it is desired to make all coincidental subsets of *S3* explicit. This can be accomplished by the command:

```
MODIFY IRD-SCHEMA-STRUCTURE S3
INCLUDE SUBSETS S2;
```

Figure B8 shows the results of this command.



**Example 9**

Using the IRD-schema-structures at the completion of the last example, suppose now that a new IRD-schema-structure *S4* is to be created as follows:

```
ADD IRD-SCHEMA-STRUCTURE S4
INCLUDE SUBSETS S1
    MEMBERS E9, R8;
```

In order to include *S4* in *S3*, the following command should be issued:

```
MODIFY IRD-SCHEMA-STRUCTURE S3
DISCONNECT S1
INCLUDE SUBSETS S4;
```

Figure B9 shows the results of this sequence of commands. Note that as a result of this command, *S1* is now a transitive subset of *S3*, rather than a direct subset of *S3*. Also note that the same result could have been obtained had *S1* been excluded from *S3*.

**Example 10**

Now suppose *S1* is modified as follows:

```
MODIFY IRD-SCHEMA-STRUCTURE S1
    INCLUDE MEMBERS E9, R9;
```

Figure B10 shows the results of this command.

Note that since *S1* is a subset of *S4*, and *S4* is a subset of *S3*, the membership of both *S4* and *S3* has been affected.

Also note that this command introduces a cycle in *S4* and *S3*. Thus neither *S4* nor *S3* could be a subset of the Phase Sensitivity Structure.

--- BEFORE COMMAND ---

(no IRD schema structure exists)

--- COMMAND ---

ADD IRD-SCHEMA-STRUCTURE S1  
INCLUDE MEMBERS E1, E2, R1;

--- AFTER COMMAND ---

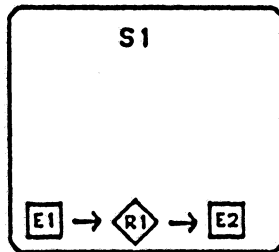
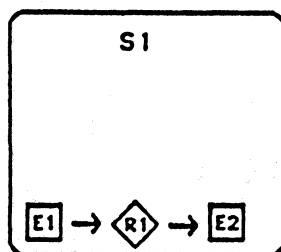


Figure B1  
Schema Structure for Example 1

--- BEFORE COMMAND ---



---COMMAND ---

ADD IRD-SCHEMA-STRUCTURE S2  
 INCLUDE MEMBERS E3, E4, E5, E6, R2, R3;

--- AFTER COMMAND ---

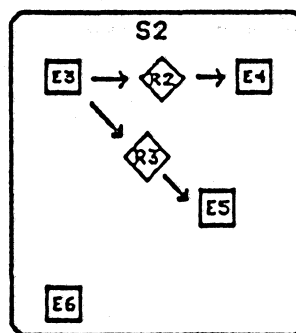
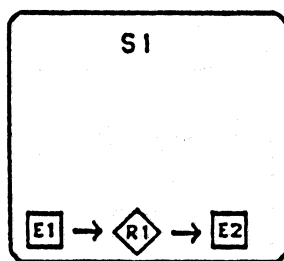
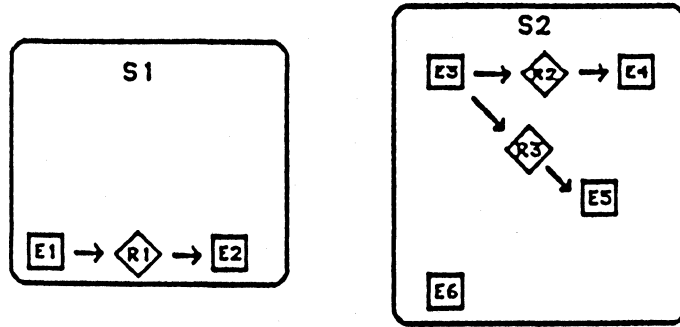


Figure B2  
 Schema Structures for Example 2

--- BEFORE COMMAND ---



---COMMAND ---

ADD IRD-SCHEMA-STRUCTURE S3  
INCLUDE  
SUBSETS S2  
APPEND MEMBERS OF S1;

--- AFTER COMMAND ---

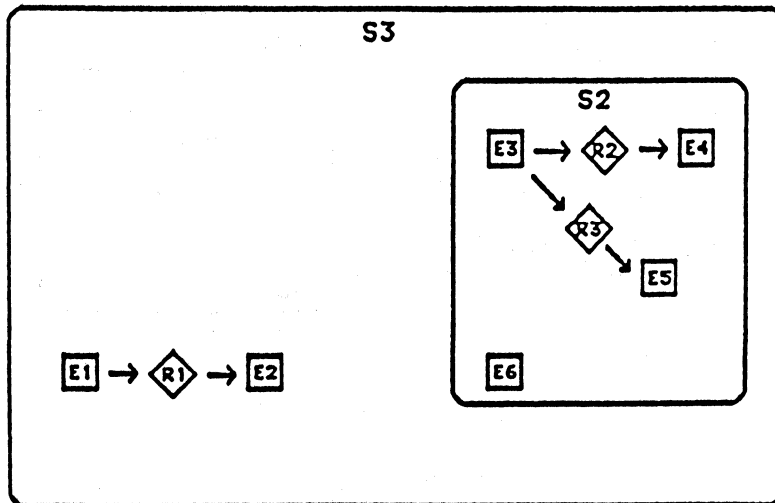
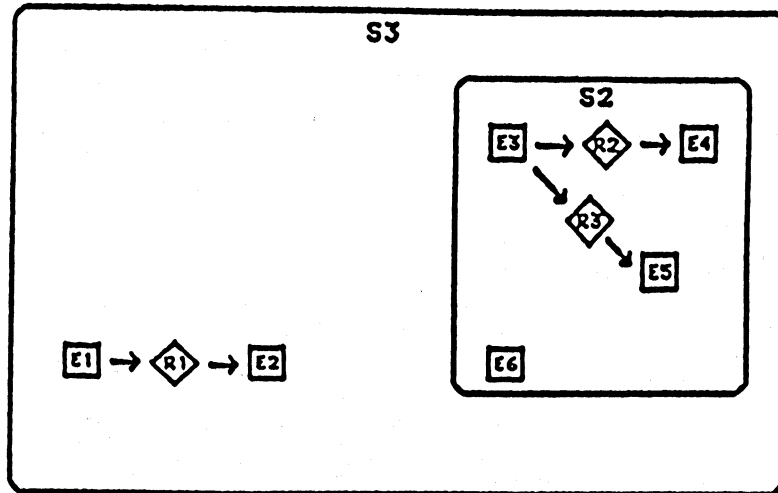


Figure B3  
Schema Structures for Example 3

--- BEFORE COMMAND ---



--- COMMAND ---

MODIFY IRD-SCHEMA-STRUCTURE S3  
 INCLUDE  
 SUBSETS S1  
 MEMBERS R4, E7, E8, R5, R6, R7;

--- AFTER COMMAND ---

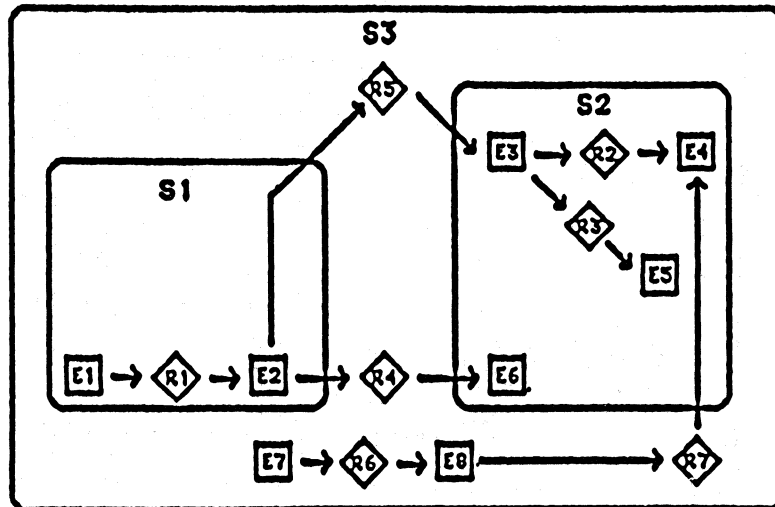
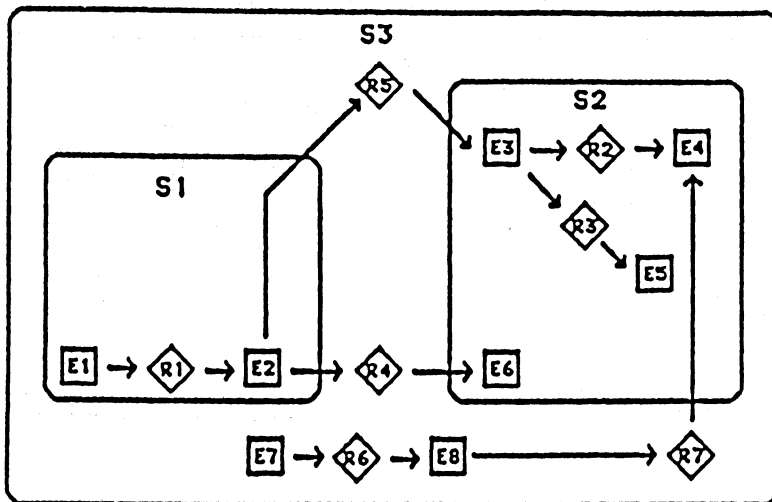


Figure B4  
 Schema Structures for Example 4

--- BEFORE COMMAND ---



---COMMAND ---

MODIFY IRD-SCHEMA-STRUCTURE S2  
EXCLUDE MEMBERS E6;

--- AFTER COMMAND ---

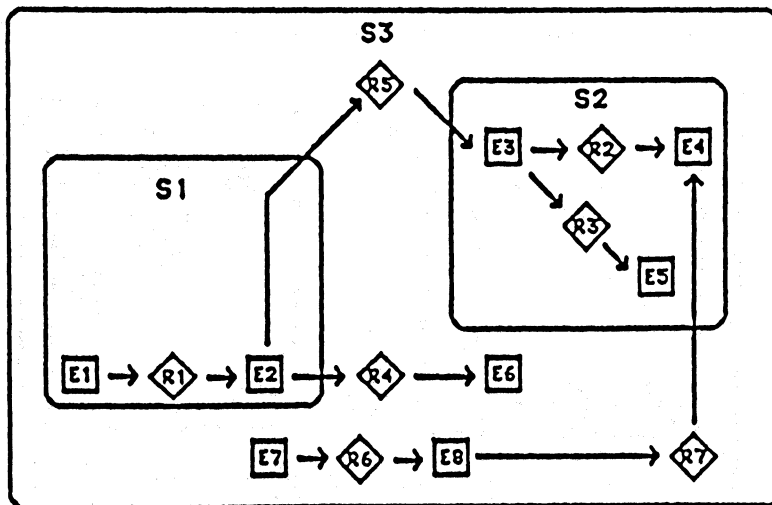
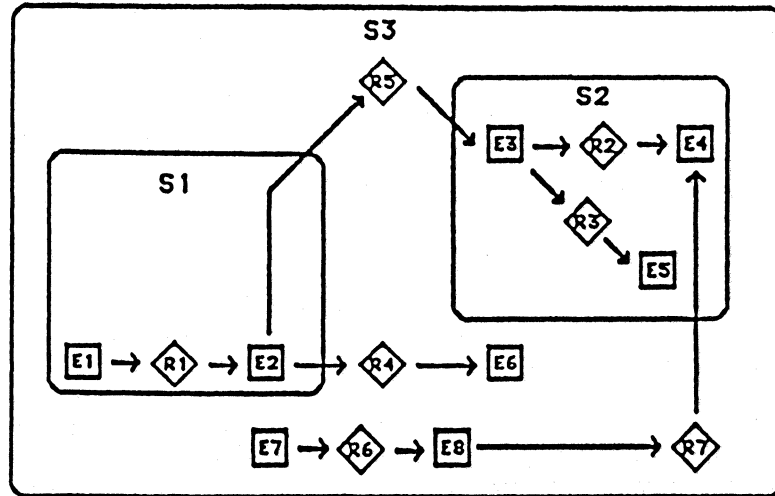


Figure B5  
Schema Structures for Example 5

--- BEFORE COMMAND ---



--- COMMAND ---

MODIFY IRD-SCHEMA-STRUCTURE S2  
INCLUDE MEMBERS E8, R7;

--- AFTER COMMAND ---

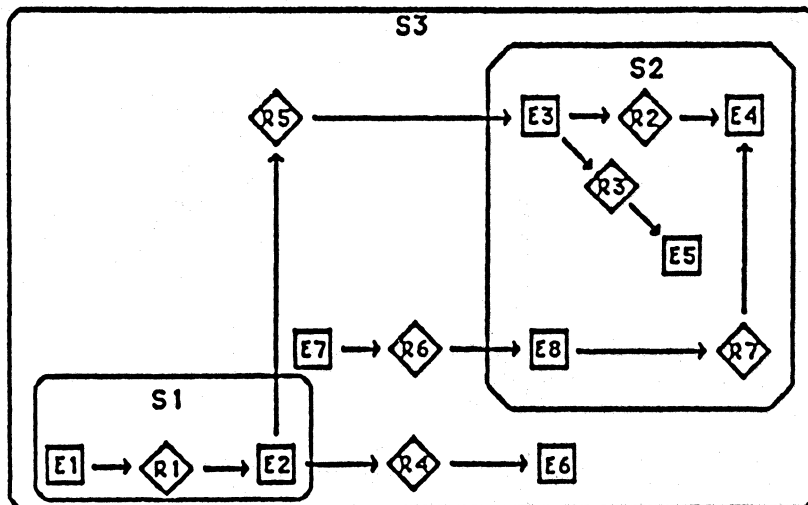
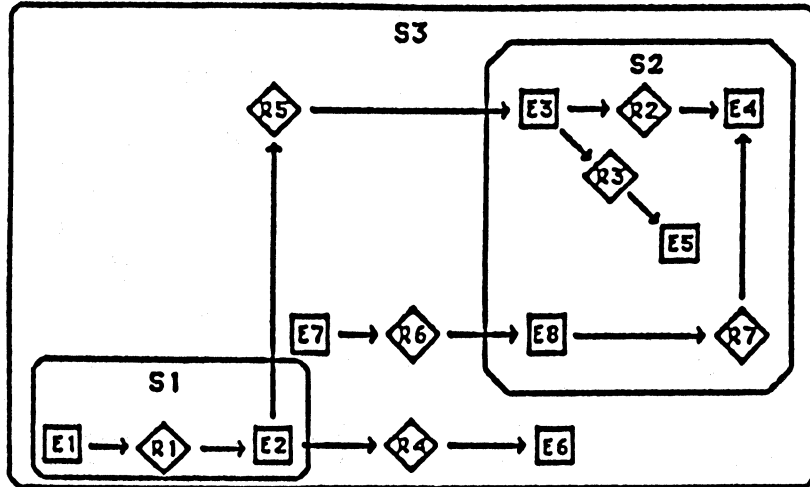


Figure B6  
Schema Structures for Example 6

--- BEFORE COMMAND ---



--- COMMAND ---

MODIFY IRD-SCHEMA-STRUCTURE S3  
 DISCONNECT S2  
 EXCLUDE MEMBERS R7;

--- AFTER COMMAND ---

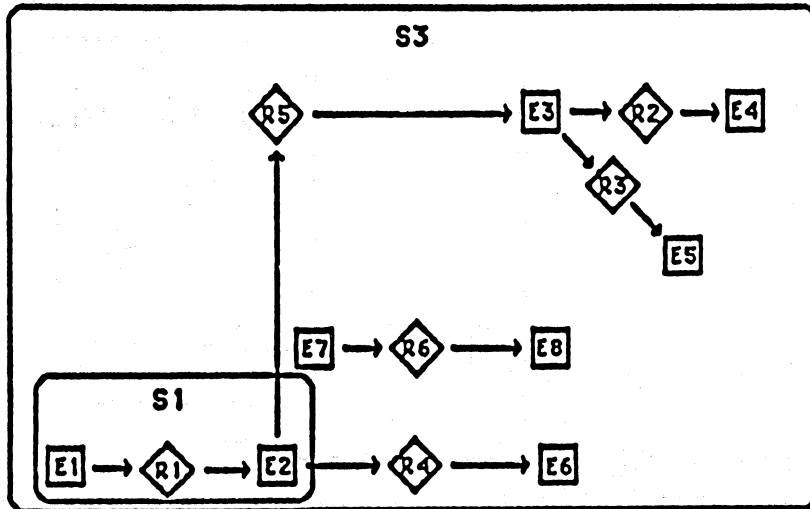
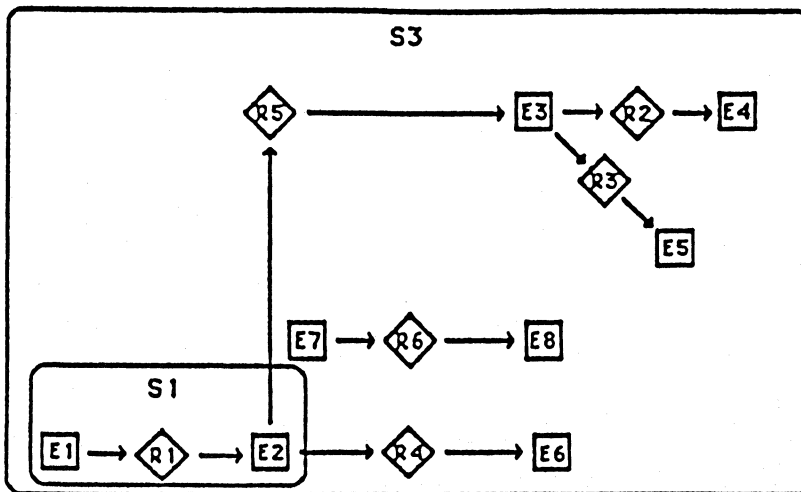


Figure B7  
 Schema Structures for Example 7



--- BEFORE COMMAND ---



--- COMMAND ---

MODIFY IRD-SCHEMA-STRUCTURE S3  
INCLUDE SUBSET S2;

--- AFTER COMMAND ---

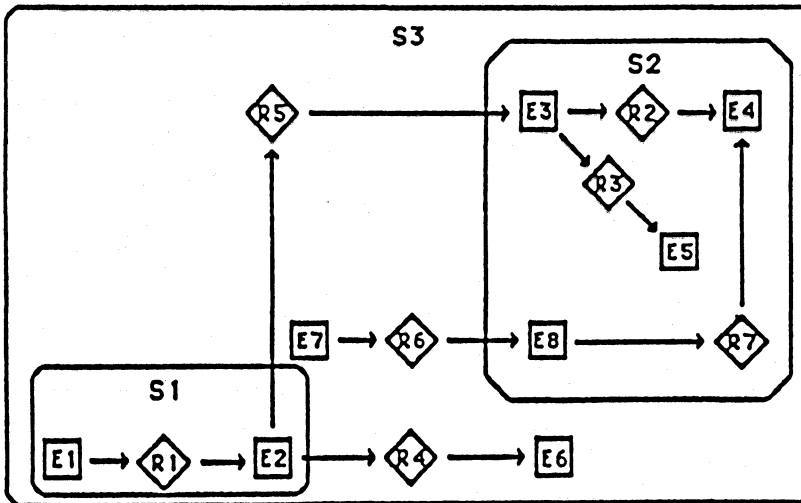


Figure B8  
Schema Structures for Example 8

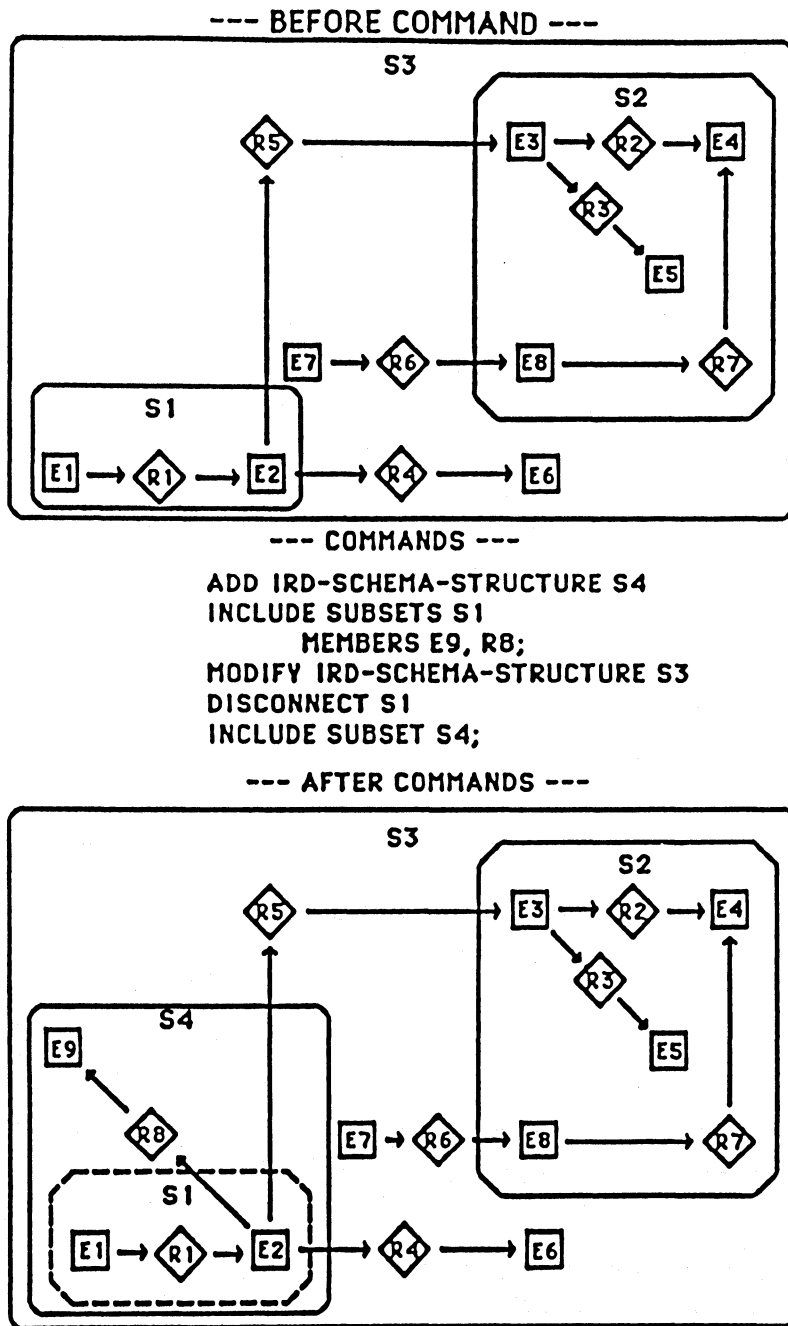
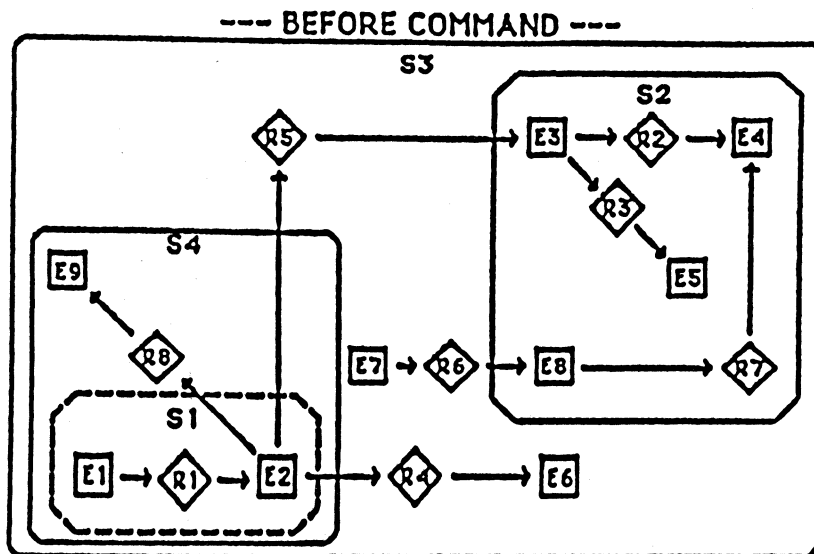


Figure B9  
Schema Structures for Example 9



--- COMMAND ---

MODIFY IRD-SCHEMA-STRUCTURE S1  
INCLUDE MEMBERS E9, R9;

--- AFTER COMMAND ---

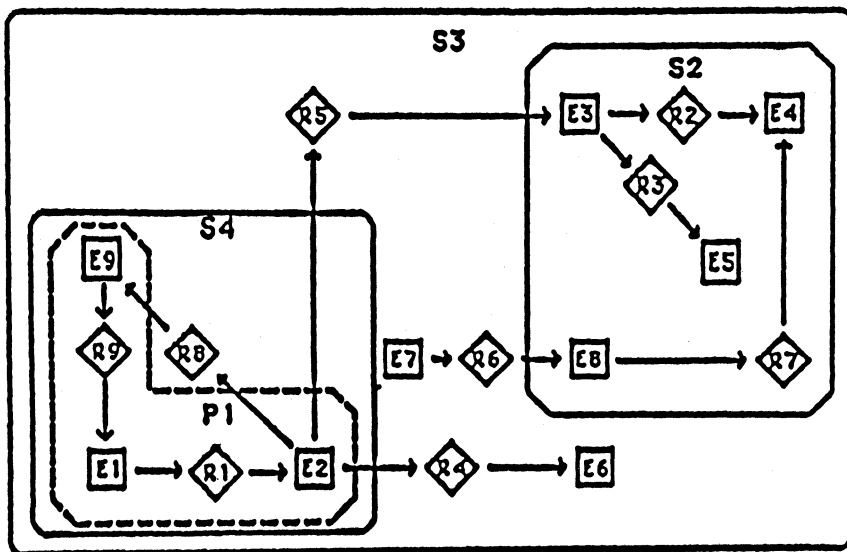


Figure B10  
Schema Structures for Example 10



## Chapter 5

### Module 5 - IRDS Procedures

#### CONTENTS

SECTION	PAGE
1 Scope, Purpose, and Application . . . . .	5-1
1.1 Scope . . . . .	5-1
1.2 Purpose . . . . .	5-1
1.3 Conformance . . . . .	5-1
1.4 Organization . . . . .	5-2
1.5 Notation . . . . .	5-3
1.6 Relationship to Other Modules . . . . .	5-3
2 References . . . . .	5-4
3 Module Overview . . . . .	5-5
4 Basic Language Elements . . . . .	5-5
5 IRDS Procedure Commands . . . . .	5-6
5.1 Run IRDS Procedure Command . . . . .	5-6
6 IRDS Procedure Command-Clauses . . . . .	5-8
6.1 Passing Arguments Command-Clause . . . . .	5-9
6.2 Return On Error Command-Clause . . . . .	5-9
6.3 Show IRDS Commands Command-Clause . . . . .	5-10
7 IRDS Procedure Language . . . . .	5-10
7.1 Procedure Statements . . . . .	5-10
7.1.1 Procedure Language Statements . . . . .	5-11
7.1.1.1 Assignment Statement . . . . .	5-11
7.1.1.2 Arg Statement . . . . .	5-12
7.1.1.3 Do Statement . . . . .	5-12
7.1.1.4 If Statement . . . . .	5-16
7.1.1.5 Case Statement . . . . .	5-17
7.1.1.6 Nop Statement . . . . .	5-19
7.1.1.7 Return Statement . . . . .	5-19
7.2 Procedure Language Expressions . . . . .	5-20
7.2.1 Terms . . . . .	5-20
7.2.2 Operators . . . . .	5-21
7.2.3 Procedure Functions . . . . .	5-22
7.2.3.1 Arg Function . . . . .	5-23
7.2.3.2 Compare Function . . . . .	5-24
7.2.3.3 Date Function . . . . .	5-25
7.2.3.4 Len Function . . . . .	5-25
7.2.3.5 Max Function . . . . .	5-26
7.2.3.6 Min Function . . . . .	5-26
7.2.3.7 Substr Function . . . . .	5-27

SECTION	PAGE
7.2.3.8 Time Function . . . . .	5-28
7.2.3.9 User Function . . . . .	5-28
7.2.3.10 View Function . . . . .	5-29
8 Modifications to Module 1 Commands and Panel Trees . . . . .	5-29
8.1 Impact on Module 1 Commands . . . . .	5-30
8.1.1 Modifications to All Commands . . . . .	5-30
8.1.2 Modifications to the Add Entity Function . . . . .	5-31
8.1.3 Modifications to the Modify Entity Function . . . . .	5-31
8.1.4 Modifications to the Modify Entity Life Cycle Phase Function . . . . .	5-31
8.1.5 Modifications to the Copy Entity Function . . . . .	5-31
8.1.6 Modifications to Export IRD Function . . . . .	5-31
8.1.7 Modifications to Check IRD Schema Compatibility Function . . . . .	5-32
8.1.8 Modifications to Import IRD Function . . . . .	5-32
8.2 Impact on Module 1 Panel Trees . . . . .	5-32
9 Modifications to the IRD Schema Definition . . . . .	5-32
9.1 New Meta-Entity of Type IRD-PARTITION . . . . .	5-33
9.2 Modification to the Meta-entity of type IRDS-RESERVED-NAMES . . . . .	5-33
9.3 New Meta-attribute-types . . . . .	5-33
9.4 Modifications to the Meta-Attribute-Type ENTITY-CLASS . . . . .	5-33
10 Extensions to the Minimal IRD Schema . . . . .	5-33
10.1 New Entity-Types . . . . .	5-34
10.2 New Relationship-Types . . . . .	5-34
10.3 New Attribute-Types . . . . .	5-35
11 IRDS Procedure Panel Trees . . . . .	5-35
11.1 Run IRDS Procedure Panel Tree . . . . .	5-35
12 Modifications to Module 2 . . . . .	5-36
13 Modifications to Module 3 . . . . .	5-36
14 Modifications to Module 4 . . . . .	5-37
15 Error and Warning Condition Usage Index . . . . .	5-37
Appendixes	
Appendix A - Collected Syntax . . . . .	5-39
Appendix B - Defining and Using IRDS Procedures . . . . .	5-43
Tables	
Table 1 - Suggested Review Sequence for this Module . . . . .	5-4

## **Chapter 5**

### **Module 5 - IRDS Procedures**

#### **1 Scope, Purpose, and Application**

##### **1.1 Scope**

Module 5 of the standard IRDS defines the means by which Information Resource Dictionary System (IRDS) procedures can be defined and executed on the IRD and IRD Schema.

##### **1.2 Purpose**

The purpose of this Module of the standard for Information Resource Dictionary Systems (IRDSs) is to provide a mechanism for defining and executing procedures on the IRDS. IRDS procedures can contain irds-commands and flow-control or assignment statements. IRDS procedures can also contain substitutable parts whose values are set at execution time.

An IRDS procedure can be used for a variety of purposes. It can, for example, be used to store a lengthy (and frequently used) irds-command for subsequent use. It can be used to store a command with the ability to alter one or more of the terms at execution time. It can be used to simplify entry of repetitive commands, or to hold the set of commands necessary to accomplish a particular task. Several examples of the use of IRDS procedures can be found in Appendix B.

An installation can use this Module as follows:

- (1) Using the IRD Schema maintenance facility, the new entity-types, relationship-types and attribute-types can be added to the IRD Schema.
- (2) Appropriate meta-relationships can be established among these meta-entities.
- (3) IRDS-PROCEDURE entities can be defined, using the IRDS IRD maintenance facility, whose DEFINITION attribute can contain the statements of the procedure. (An IRDS-PROCEDURE entity can be retained until it was deleted using the IRDS IRD maintenance facility.)
- (4) An IRDS procedure can be executed using the Run IRDS Procedure command or panel-tree.

##### **1.3 Conformance**

The Definition of a Conformant Implementation of Module 5 of a standard IRDS is given in the Definition of a Conformant Implementation of the IRDS (see Section 4 of Requirements for a Conformant Implementation).

## 1.4 Organization

Module 5 of the standard IRDS consists of the following:

- (1) A Command Language.
- (2) A Procedure Language.
- (3) Modifications to the Syntax and Rules of Module 1.
- (4) Modifications to the IRD Schema Definition.
- (5) Modifications to the Minimal IRD Schema.
- (6) Modifications to Module 2.
- (7) Modifications to Module 3.
- (8) Modifications to Module 4.

The Command Language defined in this Module is an upward compatible extension to the Command Language defined in Module 1. Wherever appropriate, the syntax components defined in Module 1 are reused.

The Procedure Language is used to define IRDS procedures. The language allows imbedded Command Language syntax as defined in Module 1 and extended in this Module.

The modifications to the syntax and rules of Module 1 take two forms:

- (1) Additional or modified rules for all commands and for specific commands.
- (2) An additional Action Area function for all panel trees.

The extensions to the IRD Schema Definition consist of a new IRD Partition and associated rules.

The extensions to the Minimal IRD Schema consist of:

- (1) An additional entity-type.
- (2) An additional relationship-type.
- (3) An additional attribute-type.

Section 3 provides an overview of where IRDS Procedures fit into the Information Resource Dictionary System, how they are defined and executed.

Section 4 defines the basic elements of the command language syntax. This section consists of new elements not defined in Module 1.

The new Commands associated with this Module are defined in Section 5. The format is the same as found in Module 1 for command specifications.

The Command-Clauses which are unique to this Module are defined in Section 6.

Section 7 describes the Language used to define IRDS procedures.

The modifications to the Command Language and Panel Trees of Module 1 are given in Section 8.

The modifications to the IRD Schema Definition are described in Section 9.

Section 10 defines the extensions to the Minimal IRD Schema which are necessary to define IRDS procedures.



The Panel Trees associated with this Module are defined in Section 11.

The modifications to Module 2 are described in Section 12.

The modifications to the Security Module (Module 3) are described in Section 13.

The modifications to the Extensible Life Cycle Phase Facility Module (Module 4) are described in Section 14.

Section 15 provides an index of the error and warning conditions defined in this Module.

The suggested order of review for this chapter is given in Table 1.

### 1.5 Notation

This Module uses the same notational conventions as identified in Module 1 of the IRDS standard, except as noted below.

The Commands and Command-Clauses defined in this Module have cross-references to syntax defined both in this Module and in other Modules. Whenever a cross reference number is given as a sequence of numbers alone, the reference is to a section or subsection within this Module. References to other Modules of this standard begin with  $Mn$ , where  $n$  denotes the number of the other Module of the IRDS standard.

Thus:

"6.2" denotes a reference to Subsection 6.2 of this Module;

whereas

"M1.4.3" is a reference to Subsection 4.3 of Module 1.

Error and warning conditions have the following format:

- (1) One alphabetic character which defines the message level:
  - (a) E identifies an error.
  - (b) W identifies a warning.
- (2) Two digits identify the Module of the standard IRDS where the message originates.
- (3) Three digits identify the message number within the Module of the standard IRDS and type of condition. (Thus error and warning conditions may have the same last three digits, i.e., E05001 and W05001 are two different conditions.)

### 1.6 Relationship to Other Modules

Additional Modules of the standard IRDS which can be developed and which have dependencies on this Module shall be required to define those dependencies within those Modules. This Module identifies its effects only on Modules 1 through 4.

**Table 1**  
**Suggested Review Sequence for this Module**

ORDER	SECTION	TITLE	ACTION
1	1	SCOPE, PURPOSE AND APPLICATION	READ
2	2 - 4	REFERENCES, MODULE OVERVIEW, BASIC LANGUAGE ELEMENTS	SKIM
3	5	IRDS PROCEDURE COMMANDS	READ
4	10	EXTENSIONS TO THE MINIMAL IRD-SCHEMAS	SKIM
5	9	MODIFICATIONS TO THE IRD SCHEMA DEFINITION	SKIM
6	7	IRDS PROCEDURE LANGUAGE	READ
7	11	IRDS PROCEDURE PANEL TREES	READ
8	8	MODIFICATIONS TO MODULE 1 COMMANDS AND PANEL TREES	READ
9	12	MODIFICATIONS TO MODULE 2	READ
10	13	MODIFICATIONS TO MODULE 3	READ
11	14	MODIFICATIONS TO MODULE 4	READ
12	6	IRDS PROCEDURE COMMAND-CLAUSES	AS REQUIRED
13	15	CROSS REFERENCE INDICES	AS REQUIRED

An implementation which provides the functionality of another Module P which depends upon this Module shall conform to this Module in order to be regarded as in conformance with Module P of the standard IRDS.

## 2 References

None.

### 3 Module Overview

This Module defines a meta-entity of type ENTITY-TYPE with assigned-meta-entity-name IRDS-PROCEDURE. Instantiations of this entity-type represent IRDS procedures.

IRDS Procedures can be executed against the Information Resource Dictionary (IRD) and/or IRD Schema using IRDS commands.

Entities of type IRDS-PROCEDURE exist only in an IRD partition named IRDS-PROCEDURE-PARTITION of the IRD.

There shall be only one entity of type IRDS-PROCEDURE with a given assigned-entity-access-name.

Entities of type IRDS-PROCEDURE are shall be created and maintained using the standard IRD-maintenance-commands described in Module 1.

Attributes of type DEFINITION associated with such entities are used to store the procedure statements corresponding to an IRDS-PROCEDURE entity; the procedure statements used to define the IRDS procedure are described in Section 7. IRDS procedures can contain irds-commands and flow-control and assignment statements. An IRDS procedure can call another IRDS procedure by embedding the irds-command which executes an IRDS procedure; IRDS procedures can thus be nested to an implementor-defined depth, which shall be at least 10. A set of built-in functions is provided to ease character and numeric manipulations and extract system information. IRDS procedures can contain substitutable parts whose values are set at execution time.

IRDS procedures are executed by issuing a Run IRDS Procedure command. Arguments can be passed to the IRDS procedure to set run-time options and values.

### 4 Basic Language Elements

**Function** To identify those additional terminal and nonterminal symbols which are provided with this Module.

**Format**

```
/* These words are additional irds-words. */
/* The presentation format is the same as for irds-words. */
```

```
arg-word ::= ARG
by-word ::= BY
case-word ::= CASE
compare-word ::= COMPARE
date-word ::= DATE
do-word ::= DO
end-word ::= END
error-word ::= ERROR
```

forever-word ::=	FOREVER
if-word ::=	IF
irds-commands-word ::=	IRDS-COMMANDS
irds-procedure-word ::=	IRDS-PROCEDURE
iterate-word ::=	ITERATE
leave-word ::=	LEAVE
len-word ::=	LEN
max-word ::=	MAX
min-word ::=	MIN
nop-word ::=	NOP
on-word ::=	ON
otherwise-word ::=	OTHERWISE
passing-word ::=	PASSING
rc-word ::=	RC
return-word ::=	RETURN
run-word ::=	RUN
substr-word ::=	SUBSTR
then-word ::=	THEN
time-word ::=	TIME
until-word ::=	UNTIL
user-word ::=	USER
view-word ::=	VIEW
when-word ::=	WHEN
while-word ::=	WHILE

**Syntax Rules**    None.

**General Rules**    None.

**Error and Warning Conditions**    None.

## 5 IRDS Procedure Commands

IRDS procedures are maintained using the IRD-maintenance-commands described in Module 1 as modified in Section 8 of this Module. The command required to execute an IRDS procedure is described in Subsection 5.1.

### 5.1 Run IRDS Procedure Command

**Function**    To execute an IRDS procedure.

**Format**

run-irds-procedure-command ::=	
run-word	4
irds-procedure-word	4
irds-procedure-name	
[ using-IRD-views-clause ]	M1.6.38
[ using-IRD-schema-views-clause ]	M1.6.132
[ passing-arguments-clause ]	6.1

[ return-on-error-clause ]	6.2
[ show-irds-commands-clause ]	6.3

irds-procedure-name ::=	
entity-access-name	M1.4.4.2

### Syntax Rules

- (1) The using-IRD-views-clause and using-IRD-schema-views-clause are only valid if Module 3 is installed.

### General Rules

- (1) IRDS-procedure-name shall be the assigned-access-name of an entity of type IRDS-PROCEDURE which exists in the IRD.

### Access Rules

The following rules apply if Module 3 is installed:

- (1) The effective IRDS-USER shall have read-permission for entities of type IRDS-PROCEDURE.
- (2) If using-IRD-views-clause is specified, it shall determine the visibility of entities and relationships that can be read by the IRDS procedure. If using-IRD-views-clause is not specified, the default IRD-view for the effective IRDS-USER shall determine the visibility of entities and relationships in the IRD.
- (3) If using-IRD-schema-views-clause is specified, it shall determine the visibility of meta-entities and meta-relationships that can be read by the IRDS procedure. If using-IRD-schema-views-clause is not specified, the default IRD-schema-view for the effective IRDS-USER shall determine the visibility of meta-entities and meta-relationships in the IRD Schema.
- (4) The IRD-view and/or IRD-schema-view in effect for writing to the IRD or IRD Schema, respectively, shall be determined by a view-function (see Subsection 7.2.3.10) contained in a procedure-statement. Any IRD-view and/or IRD-schema-view so specified shall be related to the effective IRDS-USER. Initially, these views are set to the default IRD-view and default IRD-schema-view, respectively, in effect for the IRDS-USER issuing the run-irds-procedure-command.
- (5) Any IRD-view specified in using-IRD-views-clause shall be related to the effective IRDS-USER.
- (6) If an IRDS procedure contains a Run IRDS Procedure command (embedded IRDS procedure), the IRD-view and/or IRD-schema-view in effect for the initial IRDS procedure also applies to the embedded IRDS procedure unless the embedded IRDS procedure itself contains a using-IRD-views-clause and/or using-IRD-schema-views-clause or the embedded IRDS procedure alters the effective view using the view-function (see Subsection 7.2.3.10).

### Actions

- (1) The run-irds-procedure-command shall be validated. If the run-irds-procedure-command does not pass all validations, all appropriate messages and warnings are issued.

- (2) The IRDS procedure shall be obtained and validated. Each procedure-language-statement shall be evaluated and executed, beginning with the first statement, and continuing as determined by the logic of the procedure-language-statements. If errors are encountered in a procedure-language-statement, notification shall be given to the user, and execution of the IRDS procedure shall be terminated.
- (3) If return-on-error-clause is specified in the command and if the return-code from any irds-command executed by the IRDS procedure indicates that an error was encountered, notification shall be given to the user, and execution of the IRDS procedure shall be terminated.
- (4) A run-irds-procedure-command shall issue a return-code according to the following:
  - (a) A return-code value of "ok" indicates that the IRDS procedure executed without errors.
  - (b) A return-code value of "partial" indicates that the IRDS procedure executed with errors or partially executed (e.g., a run-irds-procedure-command executed one or more embedded irds-commands, and terminated when an error was encountered).
  - (c) A return code value of "error" indicates that the IRDS procedure was not executed at all, due to errors.
- (5) If show-irds-commands-clause is specified in the command, each irds-command executed by the IRDS procedure shall be displayed prior to its execution.
- (6) Output from each irds-command executed by the IRDS procedure shall be displayed as if the command had been entered by the user.
- (7) When the logical end of the IRDS procedure has been reached, confirmation of procedure completion shall be given to the user.

#### **Error and Warning Conditions**

- (1) Error E05016: Specified IRDS procedure does not exist. See General Rule (1).
- (2) Error E01044: Invalid or unrecognized command-clause.
- (3) Error E01111: Invalid IRD-View. See Security Rule (5).

NOTE: The IRD-view or IRD-schema-view to be used by an IRDS procedure shall either be specified in a using-IRD-views-clause or using-IRD-schema-views-clause or defined within the IRDS procedure using the view-function (see Subsection 7.2.3.10).

## **6 IRDS Procedure Command-Clauses**

Three new command-clauses are required in this Module. These are described below.

## 6.1 Passing Arguments Command-Clause

**Function** To identify the arguments (parameters) to be passed to an IRDS procedure.

### Format

```

passing-arguments-clause ::=
    passing-word           4
    argument-list

argument-list ::=
    argument
    [ , argument-list ]

argument ::=
    literal                M1.4.2.1
    | access-name         M1.4.4.2

```

### Syntax Rules

- (1) The maximum number of arguments which can be passed to an IRDS procedure shall be implementation dependent. The number shall be at least 32.

### General Rules

- (1) Arguments are referenced in an arg-statement or used in the ARG function. See Subsections 7.1.1.2 and 7.2.3.1.

### Error and Warning Conditions

- (1) Error E05001: Maximum allowed number of arguments exceeded. Syntax Rule (1) has been violated.

## 6.2 Return On Error Command-Clause

**Function** To indicate whether execution of an IRDS procedure should terminate if an error is encountered in executing an imbedded irds-command.

### Format

```

return-on-error-clause ::=
    return-word           4
    on-word              4
    error-word           4

```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

### 6.3 Show IRDS Commands Command-Clause

**Function** To indicate that irds-commands executed by an IRDS procedure are to be displayed prior to their execution.

**Format**

show-irds-commands-clause ::=	
show-word	M1.4.3.2
irds-commands-word	4

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

## 7 IRDS Procedure Language

Within this Section the language used to define IRDS procedures is described. Subsection 7.1 defines the statements which are used in defining the IRDS procedure, and Subsection 7.2 describes how expressions are defined.

### 7.1 Procedure Statements

**Function** To describe the statements which comprise the definition of an IRDS procedure.

**Format**

/\* None. This breakdown is explanatory only. \*/

procedure-statement ::=	
procedure-language-statement	7.1.1
irds-command	M1.5.0.2
comment	M1.4.3.2

**Syntax Rules** None.

**General Rules** None.

**Access Rules**

- (1) If Module 3 is installed, the security rules which apply to an irds-command are applied at the time the command is executed by an IRDS procedure.

**Actions**

- (1) Each statement of the IRDS procedure shall be scanned.
  - (a) If the statement is a comment, it shall be ignored.



- (b) If the statement is a procedure-language-statement, it shall be validated. If any errors are encountered, the appropriate messages are output and execution of the IRDS procedure shall be terminated.
- (c) If the statement is not recognized as a procedure-language-statement, then it shall be assumed that the statement is an irds-command. Each term shall be evaluated as a procedure-language-expression. When the command terminator is reached the entire irds-command, including any substituted values, shall be validated and executed as described in Module 1.

**Error and Warning Conditions** None.

### 7.1.1 Procedure Language Statements

**Function** To identify the statements used to define assignments and the flow of an IRDS procedure.

**Format**

/\* None. This breakdown is explanatory only. \*/

```

procedure-language-statement ::=
    assignment-statement           7.1.1.1
    | arg-statement                 7.1.1.2
    | do-statement                  7.1.1.3
    | if-statement                  7.1.1.4
    | case-statement                7.1.1.5
    | nop-statement                 7.1.1.6
    | return-statement              7.1.1.7
    
```

**Syntax Rules** None.

**General Rules** None.

**Access Rules** None.

**Actions** None.

**Error and Warning Conditions** None.

#### 7.1.1.1 Assignment Statement

**Function** To assign values to variables.

**Format**

```

assignment-statement ::=
    variable-name           7.2.1
    =
    procedure-language-expression 7.2
    ;
    
```

**Syntax Rules** None.

**General Rules** None.

**Access Rules** None.

**Actions**

- (1) The procedure-language-expression shall be evaluated and the result assigned to variable-name.

**Error and Warning Conditions** None.

**7.1.1.2 Arg Statement**

**Function** To retrieve the values of the arguments passed to the IRDS procedure and to specify the names by which they will be identified within the IRDS procedure.

**Format**

```
arg-statement ::=
    arg-word           4
    argument-name-list
    ;

argument-name-list ::=
    argument-name     7.2.1
    [ , argument-name-list ]
```

**Syntax Rules** None.

**General Rules** None.

**Access Rules** None.

**Actions**

- (1) The values of the arguments passed to the IRDS procedure are assigned to the corresponding argument-names.

**Error and Warning Conditions** None.

**7.1.1.3 Do Statement**

**Function** The Do Statement is used to group instructions together and optionally execute them repetitively.

**Format**

```
do-statement ::=
    do-phrase
    statement-or-phrase-list
    end-phrase

do-phrase ::=
    do-word           4
```

```

    [ repetitor ]
    [ conditional ]
    ;

repetitor ::=
    control-variable
    | control-expression-1
    | forever-word 4

control-variable ::=
    control-name
    =
    control-expression-2
    [ to-expression ]
    [ by-expression ]
    [ for-expression ]

control-name ::=
    variable-name 7.2.1

to-expression ::=
    to-word
    control-expression-3 M1.4.3.2

by-expression ::=
    by-word
    control-expression-4 4

for-expression ::=
    for-word
    control-expression-5 M1.4.3.2

control-expression-1 ::=
    procedure-language-expression 7.2

control-expression-2 ::=
    procedure-language-expression 7.2

control-expression-3 ::=
    procedure-language-expression 7.2

control-expression-4 ::=
    procedure-language-expression 7.2

control-expression-5 ::=
    procedure-language-expression 7.2

conditional ::=
    while-expression
    | until-expression

while-expression ::=
    while-word
    control-expression-6 4

```

until-expression ::=	
until-word	4
control-expression-7	
control-expression-6 ::=	
procedure-language-expression	7.2
control-expression-7 ::=	
procedure-language-expression	7.2
statement-or-phrase-list ::=	
procedure-or-flow-statement	
[ statement-or-phrase-list ]	
procedure-or-flow-statement ::=	
procedure-statement	7.2
iterate-phrase	
leave-phrase	
iterate-phrase ::=	
iterate-word	4
[ control-name ]	
;	
leave-phrase ::=	
leave-word	4
[ control-name ]	
;	
end-phrase ::=	
end-word	4
[ control-name ]	
;	

**Syntax Rules**

- (1) The allowed level of nesting of do-statements shall be implementation dependent. The minimum allowed level of nesting shall be 10.

**General Rules**

- (1) The value of control-expression-2 and control-expression-4 shall be a number.
- (2) The value of control-expression-1 and control-expression-5 shall be an unsigned integer.
- (3) The value of control-expression-3, control-expression-6 and control-expression-7 shall be "true" or "false".
- (4) Statement-or-phrase-list can include any procedure-statement, including the do-statement.
- (5) If control-expression-4 is not specified, its value shall default to 1.
- (6) If control-name is specified in iterate-phrase, leave-phrase or end-phrase, it shall be the control-name of a control-variable in a currently active do-statement.

**Access Rules** None.

**Actions**

- (1) If control-name is specified in control-variable, the value of control-expression-2 shall be assigned to control-name.
- (2) The statements in statement-or-phrase-list shall be executed in the order in which they are encountered.
- (3) If end-phrase is encountered and if no repetitor or conditional was specified, execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
- (4) If end-phrase is encountered and if repetitor was control-variable
  - (a) The value assigned to control-name shall be incremented by control-expression-4.
  - (b) If to-expression is specified and the value of control-name is greater than control-expression-3, execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (c) If for-expression is specified and the statements in statement-or-phrase-list have been executed control-expression-5 times, execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (d) If while-expression is specified and control-expression-6 is "false", execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (e) If until-expression is specified and control-expression-7 is "true", execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (f) Execution of the IRDS procedure shall continue with the first statement in statement-or-phrase-list.
- (5) If end-phrase is encountered and if repetitor was control-expression-1, then
  - (a) If the statements in statement-or-phrase-list have been executed control-expression-1 times, execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (b) If while-expression is specified and control-expression-6 is "false", execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (c) If until-expression is specified and control-expression-7 is "true", execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (d) Execution of the IRDS procedure shall continue with the first statement in statement-or-phrase-list.
- (6) If end-phrase is encountered and if repetitor was forever-word or if repetitor was not specified, then
  - (a) If while-expression is specified and control-expression-6 is "false", execution of the IRDS procedure shall continue with the statement immediately following end-phrase.

- (b) If until-expression is specified and control-expression-7 is "true", execution of the IRDS procedure shall continue with the statement immediately following end-phrase.
  - (c) Execution of the IRDS procedure shall continue with the first statement in statement-or-phrase-list.
- (7) If iterate-phrase is encountered and control-name is specified,
    - (a) Execution of any active do-statements inside the one selected shall be terminated as though by a leave-phrase.
    - (b) The iterate-phrase shall act on the do-statement which includes control-name as the repetitor.
  - (8) If iterate-phrase is encountered and no control-name is specified, the iterate-phrase shall act on the innermost active do-statement.
  - (9) If iterate-phrase is encountered, processing shall continue as though an end-phrase were encountered within the appropriate do-statement; that is, with Actions (3) through (6) above.
  - (10) If leave-phrase is encountered and control-name is specified, the leave-phrase shall act on the do-statement which includes control-name as the repetitor (and on any active do-statements inside the one selected).
  - (11) If leave-phrase is encountered and no control-name is specified, the leave-phrase shall act on the innermost active do-statement.
  - (12) If leave-phrase is encountered, processing shall continue with the statement immediately following the corresponding end-phrase.

**Error and Warning Conditions**

- (1) Error E05006: Nesting of Do Statements exceeds allowed maximum. Syntax Rule (1) has been violated.
- (2) Error E05007: Value is not a number. General Rule (1) has been violated.
- (3) Error E05008: Value is not an unsigned integer. General Rule (2) has been violated.
- (4) Error E05009: Value is not "true" or "false". General Rule (3) has been violated.
- (5) Error E05010: Control-name is not associated with a currently active Do Statement. General Rule (6) has been violated.

**7.1.1.4 If Statement**

**Function** To conditionally execute a procedure-statement or group of procedure-statements.

**Format**

```
if-statement ::=  
    if-word  
    if-expression
```

then-word	4
then-instruction	
[ else-instruction ]	
if-expression ::=	
procedure-language-expression	7.2
then-instruction ::=	
procedure-statement	7.1
else-instruction ::=	
procedure-statement	7.1

### Syntax Rules

- (1) The allowed level of nesting of if-statements shall be implementation dependent. The minimum allowed level of nesting shall be 10.

### General Rules

- (1) The value of if-expression shall be "true" or "false".
- (2) The else-instruction binds to the nearest preceding if-expression. This means that if any if-expression is used as the then-instruction in an if-statement that has an else-instruction, it shall itself have an else-instruction.

**Access Rules** None.

### Actions

- (1) The if-expression shall be evaluated. If the value is "true", then the immediately following then-instruction shall be processed. If the value is "false" and an else-instruction is specified, the else-instruction shall be processed. If the value is "false" and no else-instruction is specified, execution shall continue with the next procedure-statement.

### Error and Warning Conditions

- (1) Error E05011: Nesting of If Statements exceeds allowed maximum. Syntax Rule (1) has been violated.
- (2) Error E05009: Value is not "true" or "false". General Rule (1) has been violated.
- (3) Error E05012: Unbalanced If/Then/Else expression. General Rule (2) has been violated.

#### 7.1.1.5 Case Statement

**Function** To conditionally execute one of several alternative statements.

### Format

case-statement ::=	
case-word	4
when-phrase-group	
otherwise-phrase	

end-word	4
;	
when-phrase-group ::=	
when-phrase	
[ when-phrase-group ]	
when-phrase ::=	
when-word	4
procedure-language-expression	7.2
then-word	4
procedure-statement	7.1
otherwise-phrase ::=	
otherwise-word	4
statement-list	
statement-list ::=	
procedure-statement	7.1
[ statement-list ]	

**Syntax Rules**

- (1) The allowed level of nesting of case-statements shall be implementation dependent. The minimum allowed level of nesting shall be 10.

**General Rules**

- (1) The value of each procedure-language-expression shall be "true" or "false".

**Access Rules** None.

**Actions**

- (1) Each when-phrase shall be evaluated.
  - (a) If the value of procedure-language-expression is "true", then the procedure-statement following the corresponding then-word shall be executed, after which control will pass to the statement following end-phrase. No further when-phrases within the case-statement are processed.
  - (b) If the value of procedure-language-expression is "false", then control shall pass to the next when-phrase.
- (2) If none of the procedure-language-expressions is "true", then the statement-list following otherwise-word shall be executed.

**Error and Warning Conditions**

- (1) Error E05013: Nesting of Case Statements exceeds allowed maximum. Syntax Rule (1) has been violated.
- (2) Error E05009: Value is not "true" or "false". General Rule (3) has been violated.



**7.1.1.6 Nop Statement**

**Function** The nop-statement is a procedure-statement which has no effect. It can be useful as the statement following then-word, else-word or otherwise-word in if-statements or select-statements.

**Format**

```
nop-statement ::=
    nop-word           4
    ;
```

**Syntax Rules** None.

**General Rules** None.

**Access Rules** None.

**Actions**

- (1) When the nop-statement is encountered, no action shall take place. Control shall pass to the next procedure-statement.

**Error and Warning Conditions** None.

**7.1.1.7 Return Statement**

**Function** To unconditionally terminate execution of the IRDS procedure.

**Format**

```
return-statement ::=
    return-word           4
    [ procedure-language-expression ] 7.2
    ;
```

**Syntax Rules** None.

**General Rules** None.

**Access Rules** None.

**Actions**

- (1) Execution of the IRDS procedure shall be terminated.
- (2) If procedure-language-expression is not specified, the return code (from the Run IRDS Procedure Command) shall be set to "ok".
- (3) If procedure-language-expression is specified, its value shall become the return code from the Run IRDS Procedure Command.

**Error and Warning Conditions** None.

## 7.2 Procedure Language Expressions

**Function** Define the expressions which can be used within the procedure language.

**Format**

```

procedure-language-expression ::=
    term                                7.2.1
    [ operator-term-list ]
    
```

```

operator-term-list ::=
    operator                            7.2.2
    term
    [ operator-term-list ]
    
```

**Syntax Rules** None.

**General Rules** None.

### 7.2.1 Terms

**Function** To define an elementary component of an expression.

**Format**

```

term ::=
    argument-name
    | variable-name
    | numeric-literal                M1.4.2.2
    | string-literal                M1.4.2.2
    | rc-word                        4
    | procedure-function            7.2.3
    | sub-expression
    
```

```

argument-name ::=
    irds-name                        M1.4.2.2
    
```

```

variable-name ::=
    irds-name                        M1.4.2.2
    
```

```

sub-expression ::=
    ( procedure-language-expression ) 7.2
    
```

**Syntax Rules**

- (1) The maximum level of nesting of subexpressions shall be implementor defined. The maximum shall be at least 10.
- (2) A subexpression shall begin with an open parenthesis and shall end with a corresponding close parenthesis.

## General Rules

- (1) The entire subexpression between parentheses shall be evaluated immediately when the term is required.
- (2) If term is a variable-name or argument-name, its value shall be used in evaluating the expression. If a variable has not been assigned a value it is uninitialized, and its value shall be the character(s) which comprise the name of the variable.
- (3) If rc-word is specified, the return-code value from the last executed irds-command shall be used in evaluating the expression. If no irds-command has been issued, a warning condition shall be raised.

## Error and Warning Conditions

- (1) Error E05002: Nesting of subexpressions exceeds allowed maximum. Syntax Rule (1) has been violated.
- (2) Error E01133: Missing or unbalanced parentheses. Syntax Rule (2) has been violated.
- (3) Warning W05003: No irds-command previously issued. See General Rule (3).

### 7.2.2 Operators

**Function** To define the separators which can be used between terms of expressions.

#### Format

```

operator ::=
    concatenation-character
    | arithmetic-operator           M1.4.3.2
    | relational-operator           M1.4.3.2
    | boolean-operator

concatenation-character ::=
    ||

arithmetic-operator ::=
    add
    | subtract
    | multiply
    | divide
    | integer-divide
    | remainder
    | exponent

add ::= +

subtract ::= -

multiply ::= *

divide ::= /

```

integer-divide ::= %

remainder ::= //

exponent ::= \*\*

### Syntax Rules

- (1) Operators shall have the following order of precedence: exponentiation, multiplication and division, addition and subtraction, concatenation, relational operators, boolean operators.

### General Rules

- (1) Concatenation-character shall only be used between two terms which are string-literals or whose values are string-literals.
- (2) Arithmetic-operators shall only be used between two terms which are numeric-literals or whose values are numeric-literals.
- (3) For relational-operators, if both terms involved are numeric, a numeric comparison shall be effected; otherwise, both terms are treated as character strings. For this character string comparison, leading and trailing blanks are ignored, and then the shorter string shall be padded with blanks on the right. The character comparison operation shall be case sensitive, and the exact collating order may depend on the character set used for the implementation.
- (4) Use of the integer-divide operator shall yield the integer portion of the result from dividing the first operand by the second.
- (5) Use of the remainder operator shall yield the remainder from dividing the first operand by the second.
- (6) When the sequence

term-1 operator-1 term-2 operator-2 term-3

is encountered, and operator-2 has a higher precedence than operator-1, then the sub-expression

term-2 operator-2 term-3

shall be evaluated first. The same rule shall be applied repeatedly as necessary.

### Error and Warning Conditions

- (1) Error E05015: Invalid procedure language expression. General Rule (1) or (2) has been violated.

### 7.2.3 Procedure Functions

**Function** To define the procedure-functions available within the procedure language.

**Format**

/\* Note. This breakdown is explanatory only. \*/

```

procedure-function ::=
    arg-function           7.2.3.1
    | compare-function     7.2.3.2
    | date-function        7.2.3.3
    | len-function         7.2.3.4
    | max-function         7.2.3.5
    | min-function         7.2.3.6
    | substr-function      7.2.3.7
    | time-function        7.2.3.8
    | user-function        7.2.3.9
    | view-function        7.2.3.10

```

**Syntax Rules**

(1) The view-function shall only be valid if Module 3 is installed.

**General Rules** None.

**Error and Warning Conditions** None.

**7.2.3.1 Arg Function**

**Function** To return an argument string, or information about the argument strings.

**Format**

```

arg-function ::=
    arg-word           4
    ( [ arg-phrase ] )

arg-phrase ::=
    arg-number
    [ , arg-option ]

arg-number ::=
    unsigned-integer  M1.4.2.2

arg-option ::=
    E
    | O

```

**Syntax Rules**

(1) No spaces shall appear between the components of the arg-function term.

**General Rules**

(1) If arg-number is not specified, the number of argument strings passed to the IRDS procedure shall be returned.

- (2) If arg-number is specified and arg-option is not, the arg-number-th string shall be returned. If the argument string does not exist, the null string shall be returned.
- (3) If arg-option is specified, the function tests for the existence of the arg-number-th argument string.
  - (a) If arg-option is E, the function shall return the value "true" if the arg-number-th argument exists; that is, if it was explicitly specified when the IRDS procedure was invoked. The value "false" shall be returned otherwise.
  - (b) If arg-option is O, the function shall return the value "true" if the arg-number-th argument was omitted; that is, if it was not explicitly specified when the routine was invoked. The value "false" shall be returned otherwise.

**Error and Warning Conditions**    None.

### 7.2.3.2 Compare Function

**Function**    To compare two character strings and identify the position where they diverge.

**Format**

```

compare-function ::=
    compare-word           4
    ( compare-phrase )

compare-phrase ::=
    string-1
    , string-2

string-1 ::=
    string-literal         M1.4.2.2
    | argument-name       7.2.1
    | variable-name       7.2.1

string-2 ::=
    string-literal         M1.4.2.2
    | argument-name
    | variable-name
    
```

**Syntax Rules**

- (1) No spaces shall appear between the components of the compare-function term.

**General Rules**

- (1) If string-1 or string-2 is an argument-name or variable-name, its value shall be a string literal.
- (2) If the values of string-1 and string-2 are identical, the function shall return the value 0.
- (3) If the values of string-1 and string-2 are not identical, the returned value shall be a positive integer which indicates the position of the first character that is not the same in both strings.

**Error and Warning Conditions**

- (1) Error E05004: Value is not a string literal. General Rule (1) has been violated.

**7.2.3.3 Date Function**

**Function** To return the current date.

**Format**

```

date-function ::=
    date-word           4
    ( )
    
```

**Syntax Rules**

- (1) No spaces shall appear between the components of the date-function term.

**General Rules**

- (1) The function shall return the date in the format CCYYMMDD, where:

```

CC = 2 digits, 00 through 99 inclusive, representing the century.
YY = 2 digits, 00 through 99 inclusive, representing the year.
MM = 2 digits, 01 through 12 inclusive, representing the month.
DD = 2 digits, 01 through 31 inclusive, representing the day of the
    month.
    
```

**Error and Warning Conditions** None.

**7.2.3.4 Len Function**

**Function** To return the length of a character string.

**Format**

```

len-function ::=
    len-word           4
    ( name-or-string )

name-or-string ::=
    argument-name     7.2.1
    | variable-name   7.2.1
    | string-literal  M1.4.2.2
    
```

**Syntax Rules**

- (1) No spaces shall appear between the components of the len-function term.

**General Rules**

- (1) If name-or-string is an argument-name or variable-name, its value shall be a string literal.

- (2) Len-function shall return the length of the string literal, omitting the delimiters.

### Error and Warning Conditions

- (1) Error E05004: Value is not a string literal. General Rule (1) has been violated.

### 7.2.3.5 Max Function

**Function** To return the largest number from a given list of numbers.

### Format

```
max-function ::=
    max-word                               4
    ( number-or-name-list )

number-or-name-list ::=
    number-or-name
    [ , number-or-name-list ]

number-or-name ::=
    numeric-literal                       M1.4.2.2
    | argument-name                       7.2.1
    | variable-name                       7.2.1
```

### Syntax Rules

- (1) No spaces shall appear between the components of the max-function term.

### General Rules

- (1) If number-or-name is an argument-name or a variable-name, its value shall be a numeric-literal.
- (2) The max-function shall return the largest number from the given list, that is, the number in the list which can be derived by adding a positive number or zero to any of the other numbers in the list.

### Error and Warning Conditions

- (1) Error E05014: Value is not a numeric-literal. General Rule (1) has been violated.

### 7.2.3.6 Min Function

**Function** To return the smallest number from a list of numbers.

### Format

```
min-function ::=
    min-word                               4
    ( number-or-name-list )               7.2.3.5
```



**Syntax Rules**

- (1) No spaces shall appear between the components of the min-function term.

**General Rules**

- (1) If number-or-name is an argument-name or a variable-name, its value shall be a numeric-literal.
- (2) The min-function shall return the smallest number from the given list, that is, the number in the list which can be derived by subtracting a positive number or zero from any of the other numbers in the list.

**Error and Warning Conditions**

- (1) Error E05014: Value is not a numeric-literal. General Rule (1) has been violated.

**7.2.3.7 Substr Function**

**Function** To return a substring from a given string of characters.

**Format**

```

substr-function ::=
    substr-word           4
    ( string-pos
      [ , length ] )

string-pos ::=
    name-or-string       7.2.3.4
    , position

position ::=
    unsigned-integer     M1.4.2.2

length ::=
    unsigned-integer     M1.4.2.2

```

**Syntax Rules**

- (1) No spaces shall appear between the components of the substr-function term.

**General Rules**

- (1) If name-or-string is an argument-name or variable-name, its value shall be a string literal.
- (2) The substr-function shall return the sub-string of the string literal that begins at the position-th character, and shall be of the specified length.
- (3) If position is greater than the length of the string literal then null-string shall be returned.
- (4) If length is omitted, it shall default to the rest of the string literal.

### **Error and Warning Conditions**

- (1) Error E05004: Value is not a string literal. General Rule (1) has been violated.
- (2) Warning W05005: Substring starting position is greater than the length of the string. See General Rule (3).

### **7.2.3.8 Time Function**

**Function** To return the current time.

#### **Format**

```
time-function ::=
    time-word           4
    ( )
```

#### **Syntax Rules**

- (1) No spaces shall appear between the components of the time-function term.

#### **General Rules**

- (1) The function shall return the time in the format HHMMSS, where:

HH = 2 digits, 00 through 23 inclusive, representing the hour.  
MM = 2 digits, 00 through 59 inclusive, representing the minute.  
SS = 2 digits, 00 through 59 inclusive, representing the second.

**Error and Warning Conditions** None.

### **7.2.3.9 User Function**

**Function** To return the name of the current IRDS-user.

#### **Format**

```
user-function ::=
    user-word           4
    ( )
```

#### **Syntax Rules**

- (1) No spaces shall appear between the components of the user-function term.

#### **General Rules**

- (1) The function shall return the name of the current IRDS-user.

**Error and Warning Conditions** None.

### 7.2.3.10 View Function

**Function** If Module 3 is installed, to set the effective IRD-view or IRD-schema-view to be used within an IRDS procedure for update of the IRD or IRD Schema, respectively.

#### Format

```

view-function ::=
    view-word                               4
    (IRD-word                               M1.4.3.2
     | IRD-schema-word                     M1.4.3.2
     , IRD-view-name )                     M1.6.38

```

#### Syntax Rules

- (1) The view function shall only be valid if Module 3 is installed.
- (2) No spaces shall appear between the components of the view-function term.

#### General Rules

- (1) The function shall set the appropriate view-name which shall be effective until either:
  - (a) Another view-function is effective, or
  - (b) The IRDS procedure is terminated.
- (2) When the IRDS procedure is terminated, the effective view shall return to that which was in effect before the IRDS procedure was executed.

#### Access Rules

- (1) The view-name being given shall be valid for the effective IRDS-USER.

#### Error and Warning Conditions

- (1) Error E01111: Invalid IRD-View. See Security Rule (1).

## 8 Modifications to Module 1 Commands and Panel Trees

This Section identifies those changes affected by this Module to the commands and panel trees defined in Module 1.

This Section maintains a parallel organization with the command specifications of Module 1. If a change affects all the commands within a group (e.g., "IRD Output Commands"), then the change shall be identified only once for the group. If a command is not explicitly identified, or it is not part of a group of commands for which changes have been identified, then there are no changes to that command.

For each affected command, only the modifications shall be identified. These can include:

- (1) Format
- (2) Syntax Rules
- (3) General Rules
- (4) Actions
- (5) Error and Warning Conditions

Unless otherwise stated, the Security Rules for each command or group of commands completely replace the corresponding Rules in Module 1.

The only modification to the panel trees are the modifications to the data area which correspond to the modifications in the command language syntax. These are identified in detail in Subsection 8.2.

### **Notation for Identifying Modifications**

Changes are identified as being in one of the following classes:

- (1) *Additions.* The additional wording or syntax does not require any changes to any other rules and/or actions. Additional syntax will be identified by a comment identifying the new syntax. Additional rules, actions, or error or warning conditions are identified by a "(+digit)" to indicate the relative position of the new text in the series.
- (2) *Modifications.* Syntax or a given rule and/or action has been changed. The overall sense of the rule or action, however, remains the same. For syntax, the change is minor.
- (3) *Replacements.* A given rule or the format has been substantially changed. The new rules are identified using the notation "(digit)" rather than the "(+digit)" notation for additions.
- (4) *Deletions.* A rule, action or syntax has been deleted. Any corresponding error/warning condition shall be also identified.

For error and warning conditions which correspond to additional rules the correspondence to the additional rule is identified using the relative number of the additional rule (i.e., "(+digit)").

### **8.1 Impact on Module 1 Commands**

This Module requires a change to all Module 1 commands and changes to several specific commands. These are described below.

#### **8.1.1 Modifications to All Commands**

##### **Actions**

- (+1) Each irds-command shall set a return code which indicates whether or not the command executed successfully as follows:
  - (a) A return code value of "ok" indicates that the command executed without errors.
  - (b) A return code value of "warn" indicates that the command executed without errors, but that a warning condition was raised.
  - (c) A return code value of "error" indicates that the command was not executed, due to errors.

### **8.1.2 Modifications to the Add Entity Function**

#### **Syntax Rules**

- (+1) If the entity-type identified in entity-type-clause is an IRDS-PROCEDURE then no version-identifier shall be specified in new-entity-name.

#### **General Rules**

- (+1) If the entity-type specified is IRDS-PROCEDURE, then the IRD-VIEW which is in effect shall be associated with the IRD-PARTITION named IRDS-PROCEDURE-PARTITION.
- (+2) If the IRD-VIEW which is in effect is associated with the IRD-PARTITION named IRDS-PROCEDURE-PARTITION, then only entities of type IRDS-PROCEDURE shall be added.

### **8.1.3 Modifications to the Modify Entity Function**

#### **General Rules**

- (+1) New-version-clause shall not be specified if the entity-type of the specified entity is an IRDS-PROCEDURE.

### **8.1.4 Modifications to the Modify Entity Life Cycle Phase Function**

#### **General Rules**

- (+1) Neither Life-Cycle-Phase specified in this command shall be IRDS-PROCEDURE-PARTITION.

### **8.1.5 Modifications to the Copy Entity Function**

#### **General Rules**

- (+1) New-version-clause shall not be specified if the existing-entity-access-name is an IRDS-PROCEDURE.

#### **Security Rules**

- (+1) If an existing-entity-access-names identifies an IRDS-PROCEDURE, then the IRD-VIEW in effect shall be associated with the IRD-PARTITION named IRDS-PROCEDURE-PARTITION.

### **8.1.6 Modifications to Export IRD Function**

#### **Actions**

- (+1) All entities exported from the IRDS-PROCEDURE-PARTITION will be clustered together at the beginning of the IRD-export-file.

### **8.1.7 Modifications to Check IRD Schema Compatibility Function**

#### **General Rules**

- (+1) If the source-IRD Schema has an IRD-PARTITION named IRDS-PROCEDURE-PARTITION, then the target-IRD Schema shall also have an IRD-PARTITION named IRDS-PROCEDURE-PARTITION.
- (+2) The meta-attributes defining implementor limits for the maximum nesting depths of IRDS procedures and subexpressions in the source IRD Schema shall be compared to these meta-attributes in the target IRD Schema. If either of these meta-attributes in the source IRD Schema is greater than the corresponding meta-attribute in the target IRD Schema, a warning shall be raised.

### **8.1.8 Modifications to Import IRD Function**

#### **Actions**

- (+1) If the target IRD Schema has an IRD-PARTITION named IRDS-PROCEDURE-PARTITION, any IRDS-PROCEDURE entities found at the beginning of the specified IRD-export-file will be imported into the IRDS-PROCEDURE-PARTITION.
- (+2) If the target IRD Schema does not have an IRD-PARTITION named IRDS-PROCEDURE-PARTITION, any IRDS-PROCEDURE entities found at the beginning of the specified IRD-export-file will be ignored by the command, and a warning will be raised and a list of the access-names of the IRDS-PROCEDURE entities contained in the IRD-export-file will be displayed.
- (+3) Any entities of type IRDS-PROCEDURE on the IRD-export-file that have an identical entity-access-name to an entity of type IRDS-PROCEDURE already in the target IRD shall be rejected and a warning will be raised and a list of the access-names of the IRDS-PROCEDURE entities contained in the IRD-export file will be displayed.

## **8.2 Impact on Module 1 Panel Trees**

#### **Actions**

- (+1) The Action Area of each Panel Tree defined in Module 1 shall include the following:

For at least one panel within each panel tree, the panel area shall contain the RUN IRDS-PROCEDURE function.

Selection of the RUN IRDS-PROCEDURE function by the user indicates the desire to run an IRDS procedure. Control shall be transferred to the Run IRDS Procedure Panel Tree. [Note that control can be returned to the current panel by using the RETURN option in the Action Area of the Run IRDS Procedure Panel Tree.]

## **9 Modifications to the IRD Schema Definition**

The following modifications to the definition of the IRD schema are required:

### 9.1 New Meta-Entity of Type IRD-PARTITION

The IRDS Procedure Module requires the presence of a meta-entity of type IRD-PARTITION named IRDS-PROCEDURE-PARTITION to which the following rules apply:

- (1) All rules in Module 1 which apply to the SECURITY partition, except those which apply to the Export IRD Command and Import IRD Command, shall also apply to the IRDS-PROCEDURE-PARTITION.
- (2) All entities of type IRDS-PROCEDURE shall reside in the IRDS-PROCEDURE-PARTITION.
- (3) The rules in Module 1 which pertain to the SECURITY partition entity-types and relationship-types shall pertain to the entity-type IRDS-PROCEDURE and the relationship-type IRDS-PROCEDURE-RUNS-IRDS-PROCEDURE.

### 9.2 Modification to the Meta-entity of type IRDS-RESERVED-NAMES

The repeating meta-attribute-type IRDS-RESERVED-NAMES associated with the meta-entity STANDARD-RESERVED-NAMES shall be modified to have additional valid values IRDS-PROCEDURE, ARG, DO, IF, CASE, NOP and RETURN.

### 9.3 New Meta-attribute-types

The following new meta-attribute-types shall be assigned to the meta-entity-type IRDS-LIMITS:

#### IRDS-PROCEDURE-NESTING-DEPTH-LIMIT

This meta-attribute states the maximum depth to which IRDS procedures can be nested, which has been set by an implementor of this Module. This limit shall not be less than 10.

#### SUBEXPRESSION-NESTING-DEPTH-LIMIT

This meta-attribute states the maximum depth to which sub-expressions can be nested, which has been set by an implementor of this Module. This limit shall not be less than 10.

### 9.4 Modifications to the Meta-Attribute-Type ENTITY-CLASS

- (1) The Maximum Length of the ENTITY-CLASS meta-attribute-type shall be changed to 14.
- (2) The Valid Values of the ENTITY-CLASS meta-attribute-type shall include IRDS-PROCEDURE.

## 10 Extensions to the Minimal IRD Schema

This subsection takes the form of IRD Schema commands. These IRD Schema commands identify the extensions to the Minimal IRD Schema defined in Module 1 which are required by the IRDS Procedure Module.

The commands are divided into the following subsections:

- (1) New entity-types.
- (2) New relationship-types.
- (3) New attribute-types.

Note that although the form of IRD Schema commands is used to describe the required changes, an implementation of the IRDS need not use this approach for installation of the IRDS Procedure Module. Likewise, these commands do not specify the installation of the meta-entities to support this Module.

### 10.1 New Entity-Types

/\* The following additional entity-type shall be defined to support IRDS Procedures. \*/

```
ADD META-ENTITY IRDS-PROCEDURE
  META-ENTITY-TYPE = ENTITY-TYPE
  WITH META-ATTRIBUTES
  META-ENTITY-SUBSTITUTE-NAME = IRDS-PROC
  CONNECTABLE = NO
  ENTITY-CLASS = IRDS-PROCEDURE
  SYSTEM-LOCK = ON
  PURPOSE =
  "This entity-type is required for the IRDS Procedure Module."
  ;
```

### 10.2 New Relationship-Types

/\* The following additional relationship-types shall be defined to support IRDS Procedures. \*/

```
ADD META-ENTITY RUNS
  META-ENTITY-TYPE = RELATIONSHIP-CLASS-TYPE
  WITH META-ATTRIBUTES
  SYSTEM-LOCK = ON
  INVERSE-NAME = RUN-BY
  PURPOSE =
  "This relationship-class-type identifies all relationship-types used
  to support the IRDS Procedure Module."
  ;
```

```
ADD META-ENTITY IRDS-PROCEDURE-RUNS-IRDS-PROCEDURE
  META-ENTITY-TYPE = RELATIONSHIP-TYPE
  WITH META-ATTRIBUTES
  META-ENTITY-SUBSTITUTE-NAME = IRDS-PROC-RUNS-IRDS-PROC
  INVERSE-NAME = IRDS-PROCEDURE-RUN-BY-IRDS-PROCEDURE
  SYSTEM-LOCK = ON
  PURPOSE =
  "This entity-type is required for the IRDS Procedure Module."
  ;
```

/\* The following IRD Schema commands establish the relationship-types needed to support IRDS Procedures. \*/



```

ADD META-RELATIONSHIP
  FROM IRDS-PROCEDURE-RUNS-IRDS-PROCEDURE TO RUNS ;

ADD META-RELATIONSHIP
  FROM IRDS-PROCEDURE-RUNS-IRDS-PROCEDURE TO IRDS-PROCEDURE
  WITH POSITION = 1 ;

ADD META-RELATIONSHIP
  FROM IRDS-PROCEDURE-RUNS-IRDS-PROCEDURE TO IRDS-PROCEDURE
  WITH POSITION = 2 ;

```

### 10.3 New Attribute-Types

/\* The following additional attribute-type shall be defined to support IRDS Procedures. \*/

```

ADD META-ENTITY DEFINITION
  META-ENTITY-TYPE = ATTRIBUTE-TYPE
  WITH META-ATTRIBUTES
  FORMAT = TEXT
  SYSTEM-LOCK = ON
  PURPOSE =
  "This attribute-type contains the procedure-statements which define
  an IRDS Procedure. The format of the procedure-statements and their
  effects are described in Subsection 7 of the IRDS Procedure Module."

```

/\*The following command links the Definition attribute-type to the IRDS Procedure entity-type.  
\*/

```

ADD META-RELATIONSHIP
  FROM IRDS-PROCEDURE TO DEFINITION
  WITH META-ATTRIBUTES
  SINGULAR = YES ;

```

## 11 IRDS Procedure Panel Trees

IRDS procedures can be maintained using the IRD maintenance panel trees described in Module 1. The panel-tree which is used to execute an IRDS procedure is described in Subsection 11.1.

### 11.1 Run IRDS Procedure Panel Tree

**Function** This panel tree is used to execute an IRDS procedure.

#### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the IRDS procedure to be executed.

- (b) If Module 3 is installed, optionally, the views to be used within the IRDS procedure. If no views are specified then the effective view will be used.
- (c) The arguments to be passed to the IRDS procedure, if any.
- (d) An indication of whether execution of the IRDS procedure is to be continued or terminated should an error be encountered in processing an irds-command.
- (e) An indication of whether irds-commands should be displayed prior to their execution by the IRDS procedure.

**IRD Schema Area Contents**     None

**Action Area Contents**

- (1) In addition to those global options for the Action Area specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Run the specified IRDS procedure, in which case the specified procedure is executed.
  - (b) "RETURN" to the panel from which control was transferred to the Run IRDS Procedure panel tree.

**12 Modifications to Module 2**

None.

**13 Modifications to Module 3**

Modifications to Module 3 of the standard IRDS, "IRDS Security", are discussed in this Section.

- (1) The read-lock is extended to apply to the ability to execute an IRDS Procedure.

/\* The following changes to attribute-types specified in Module 3 for the Security Module are required to support the IRDS Procedure Module. \*/

```
MODIFY META-ENTITY READ-PERMISSION
  WITH META-ATTRIBUTES
  PURPOSE =
```

```
"To permit a designated IRDS-USER the use of the following commands:
  All entity-list commands
  All output commands
  Run-procedure command
```

```
Valid values for this attribute-type are 'YES' and 'NO'. "
```

;

## 14 Modifications to Module 4

The following two modifications shall be made:

- (1) When the IRD is activated with this Module installed in addition to Modules 1 and 4:

DICTIONARY-STRUCTURE(1) shall also have as members the entity-type IRDS-PROCEDURE and the relationship-type IRDS-PROCEDURE-RUNS-IRDS-PROCEDURE.

/\* PHASE-SENSITIVITY-STRUCTURE(1) shall have no additional member. \*/

- (2) The rules for IRD Schema life-cycle-phase management of meta-entities in the IRD Partition named SECURITY shall also apply to meta-entities in the IRDS-PROCEDURE-PARTITION.

## 15 Error and Warning Condition Usage Index

A usage index for the error and warning conditions follows. It identifies where each error and warning condition is used in this Module. The scope of the index is this Module only.

<u>Cond</u>	<u>Text / Subsection</u>
E01044	Invalid, missing or misplaced command-clause. 5.1
E01111	Invalid IRD-View. 5.1            7.2.3.10
E01133	Missing or unbalanced parentheses. 7.2.1
E05001	Maximum number of arguments exceeded. 6.1
E05002	Nesting of sub-expressions exceeds allowed maximum. 7.2.1
E05004	Value is not a string literal. 7.2.3.2    7.2.3.4    7.2.3.7
E05006	Nesting of Do Statements exceeds allowed maximum. 7.1.1.3
E05007	Value is not a number. 7.1.1.3
E05008	Value is not an unsigned integer. 7.1.1.3

- E05009 Value is not "true" or "false".  
7.1.1.3 7.1.1.4 7.1.1.5
- E05010 Control-name is not associated with a currently active Do Statement.  
7.1.1.3
- E05011 Nesting of If Statements exceeds allowed maximum.  
7.1.1.4
- E05012 Unbalanced If/Then/Else expression.  
7.1.1.4
- E05013 Nesting of Case Statements exceeds allowed maximum.  
7.1.1.5
- E05014 Value is not a numeric-literal.  
7.2.3.5 7.2.3.6
- E05015 Invalid procedure language expression.  
7.2.2
- E05016 Specified IRDS procedure does not exist.  
5.1
- W05003 No irds-command previously issued.  
7.2.1
- W05005 Substring starting position is greater than the length of the string.  
7.2.3.7

# Appendixes

(These Appendixes are not part of American National Standard X3.138-1988, but are included for information only.)

## Appendix A

### Collected Syntax

```
add ::= +

argument ::=
    literal
    | access-name

argument-list ::=
    argument
    [ , argument-list ]

argument-name ::= irds-name

argument-name-list ::=
    argument-name
    [ , argument-name-list ]

arg-function ::=
    arg-word
    ( [ arg-phrase ] )

arg-number ::= unsigned-integer

arg-option ::=
    E
    | 0

arg-phrase ::=
    arg-number
    [ , arg-option ]

arg-statement ::=
    arg-word
    argument-name-list
    ;

arg-word ::= ARG

arithmetic-operator ::=
    add
    | subtract
    | multiply
    | divide
    | integer-divide
    | remainder
    | exponent

assignment-statement ::=
    variable-name
    =
    procedure-language-expression
    ;

by-expression ::=
    by-word
    control-expression-4

by-word ::= BY

case-statement ::=
    case-word
    when-phrase-group
    otherwise-phrase
    end-word
    ;

case-word ::= CASE

compare-function ::=
    compare-word
    ( compare-phrase )

compare-phrase ::=
    string-1
    , string-2

compare-word ::= COMPARE

concatenation-character ::= |
```

## APPENDIX

```

conditional ::=
    while-expression
  | until-expression

control-expression-1 ::=
    procedure-language-expression

control-expression-2 ::=
    procedure-language-expression

control-expression-3 ::=
    procedure-language-expression

control-expression-4 ::=
    procedure-language-expression

control-expression-5 ::=
    procedure-language-expression

control-expression-6 ::=
    procedure-language-expression

control-expression-7 ::=
    procedure-language-expression

control-name ::= variable-name

control-variable ::=
    control-name
  =
    control-expression-2
  [ to-expression ]
  [ by-expression ]
  [ for-expression ]

date-function ::=
    date-word
  ( )

date-word ::= DATE

divide ::= /

do-phrase ::=
    do-word
  [ repetitor ]
  [ conditional ]
  ;

do-statement ::=
    do-phrase
  statement-or-phrase-list
  end-phrase

do-word ::= DO

else-instruction ::= procedure-statement

end-phrase ::=
    end-word
  [ control-name ]
  ;

end-word ::= END

error-word ::= ERROR

exponent ::= **

forever-word ::= FOREVER

for-expression ::=
    for-word
  control-expression-5

if-expression ::=
    procedure-language-expression

if-statement ::=
    if-word
  if-expression
  then-word
  then-instruction
  [ else-instruction ]

if-word ::= IF

integer-divide ::= %

irds-commands-word ::= IRDS-COMMANDS

irds-procedure-name ::= entity-access-name

irds-procedure-word ::= IRDS-PROCEDURE

iterate-phrase ::=
    iterate-word
  [ control-name ]
  ;

iterate-word ::= ITERATE

leave-phrase ::=
    leave-word
  [ control-name ]
  ;

leave-word ::= LEAVE

```

```

length ::= unsigned-integer

len-function ::=
    len-word
    ( name-or-string )

len-word ::= LEN

max-function ::=
    max-word
    ( number-or-name-list )

max-word ::= MAX

min-function ::=
    min-word
    ( number-or-name-list )

min-word ::= MIN

multiply ::= *

name-or-string ::=
    argument-name
    | variable-name
    | string-literal

nop-statement ::=
    nop-word
    ;

nop-word ::= NOP

number-or-name ::=
    numeric-literal
    | argument-name
    | variable-name

number-or-name-list ::=
    number-or-name
    [ , number-or-name-list ]

on-word ::= ON

operator ::=
    concatenation-character
    | arithmetic-operator
    | relational-operator
    | boolean-operator

operator-term-list ::=
    operator
    term
    [ operator-term-list ]

otherwise-phrase ::=
    otherwise-word
    statement-list

otherwise-word ::= OTHERWISE

passing-arguments-clause ::=
    passing-word
    argument-list

passing-word ::= PASSING

position ::= unsigned-integer

procedure-function ::=
    arg-function
    | compare-function
    | date-function
    | len-function
    | max-function
    | min-function
    | substr-function
    | time-function
    | user-function
    | view-function

procedure-language-expression ::=
    term
    [ operator-term-list ]

procedure-language-statement ::=
    assignment-statement
    | arg-statement
    | do-statement
    | if-statement
    | case-statement
    | nop-statement
    | return-statement

procedure-or-flow-statement ::=
    procedure-statement
    | iterate-phrase
    | leave-phrase

procedure-statement ::=
    procedure-language-statement
    | irds-command
    | comment

rc-word ::= RC

remainder ::= //

```

## APPENDIX

```

repetitor ::=
    control-variable
    | control-expression-1
    | forever-word

return-on-error-clause ::=
    return-word
    on-word
    error-word

return-statement ::=
    return-word
    [ procedure-language-expression ]
    ;
return-word ::= RETURN

run-irds-procedure-command ::=
    run-word
    irds-procedure-word
    irds-procedure-name
    [ using-IRD-views-clause ]
    [ using-IRD-schema-views-clause ]
    [ passing-arguments-clause ]
    [ return-on-error-clause ]
    [ show-irds-commands-clause ]

run-word ::= RUN

show-irds-commands-clause ::=
    show-word
    irds-commands-word

statement-list ::=
    procedure-statement
    [ statement-list ]

statement-or-phrase-list ::=
    procedure-or-flow-statement
    [ statement-or-phrase-list ]

string-pos ::=
    name-or-string
    , position

string-1 ::=
    string-literal
    | argument-name
    | variable-name

string-2 ::=
    string-literal
    | argument-name
    | variable-name

substr-function ::=
    substr-word
    ( string-pos
      [ , length ] )

substr-word ::= SUBSTR

subtract ::= -

sub-expression ::=
    ( procedure-language-expression )

term ::=
    argument-name
    | variable-name
    | numeric-literal
    | string-literal
    | rc-word
    | procedure-function
    | sub-expression

then-instruction ::= procedure-statement

then-word ::= THEN

time-function ::=
    time-word
    ( )

time-word ::= TIME

to-expression ::=
    to-word
    control-expression-3

until-expression ::=
    until-word
    control-expression-7

until-word ::= UNTIL

user-function ::=
    user-word
    ( )

user-word ::= USER

variable-name ::= irds-name

view-function ::=
    view-word
    (IRD-word
     |IRD-schema-word
     , IRD-view-name )

```



view-word ::= VIEW

when-word ::= WHEN

when-phrase ::=

when-word  
procedure-language-expression  
then-word  
procedure-statement

while-expression ::=

while-word  
control-expression-6

while-word ::= WHILE

when-phrase-group ::=

when-phrase  
[ when-phrase-group ]

## Appendix B

### Defining and Using IRDS Procedures

The following examples of IRDS Procedures and the results when executed using the Run IRDS Procedure Command are provided for illustrative purposes only.

**Example 1.** An IRDS Procedure can be used to store a single IRDS command for subsequent use.

Defining the procedure:

```
ADD ENTITY P-Sec
  ENTITY-TYPE = IRDS-PROCEDURE
  DESCRIPTIVE-NAME = Output Security-Related Entities
  WITH ATTRIBUTES
  DESCRIPTION =
  "Procedure to output names and descriptions of security-related
  entities."
  DEFINITION =
  " OUTPUT IRD SELECT ALL ENTITIES WHERE
    DESCRIPTION = "*"security*" OR
    DESCRIPTION = "*"password*"
    SORT SEQUENCE = ENTITY-TYPE, ACCESS-NAME
    SHOW ACCESS-NAME
    SHOW ATTRIBUTE DESCRIPTION ;
    RETURN ; " ;
```

Executing the procedure:

```
RUN IRDS-PROCEDURE P-Sec USING IRD-VIEW = Sys-View;
```

**Example 2.** An IRDS procedure can be called passing parameters to set run-time options.

Defining the procedure:

## APPENDIX

```
ADD ENTITY Out-T-Elem
  ENTITY-TYPE = IRDS-PROCEDURE
  DESCRIPTIVE-NAME = Output Test Elements
  WITH ATTRIBUTES
  DESCRIPTION =
    "Procedure to output highest or specified revision of elements
    in Test Phase."
  DEFINITION =
    " VIEW(IRD,Div-101)
    /* Set revision number to passed value */
    REVNO = ARG(1);
    /* If no passed value, use highest revision */
    IF ARG(1,0) THEN REVNO = HIGHEST;
    OUTPUT IRD SELECT ALL ENTITIES WHERE
      ENTITY-TYPE = ELEMENT
      AND REVISION = REVNO
      AND PHASE = TEST
      SHOW ALL;
      RETURN; " ;
```

Executing the procedure:

```
RUN IRDS-PROCEDURE Out-T-Elem PASSING 5;
```

would be equivalent to executing the command

```
OUTPUT IRD SELECT ALL ENTITIES WHERE
  ENTITY-TYPE = ELEMENT
  AND REVISION = 5
  AND PHASE = TEST
  SHOW ALL;
```

**Example 3.** IRDS procedures can contain multiple commands and call other IRDS procedures. The first procedure below builds an entity-list. The second deletes the entities in the list.

Defining the procedures:

```
ADD ENTITY DB-Old-Proc
  ENTITY-TYPE = IRDS-PROCEDURE
  DESCRIPTIVE-NAME = Old Databases List
  WITH ATTRIBUTES
  DESCRIPTION =
    "Procedure to obtain a list of entities which are
    database-related and are no longer used."
  DEFINITION =
    " VIEW(IRD,Db-03)
    BUILD ENTITY-LIST
    SELECT ENTITIES WITH ACCESS-NAME = *ASCAD*
    WHERE (DESCRIPTION = ""*database*"" AND
      NO RELATIONSHIPS EXIST) OR
      (REVISION < HIGHEST AND PHASE = TEST)
```

```
LIST-NAME = DB-Old-List ;
RETURN ; " ;
```

```
ADD ENTITY Del-Old-DB
  ENTITY-TYPE = IRDS-PROCEDURE
  DESCRIPTIVE-NAME = Delete Old Databases
  WITH ATTRIBUTES
  DESCRIPTION =
    "Procedure to delete entities which are database-related
    and are no longer used."
  DEFINITION =
    " VIEW(IRD,Db-03)
    RUN IRDS-PROCEDURE DB-Old-Proc ;
    DELETE ENTITY USING ENTITY-LIST = DB-Old-List ;
    RETURN ; " ;
```

and the relationship

```
ADD RELATIONSHIP Del-Old-DB IRDS-PROCEDURE-RUNS-IRDS-PROCEDURE DB-
Old-Proc;
```

Executing the procedure:

```
RUN IRDS-PROCEDURE Del-Old-DB
  RETURN ON ERROR ;
```

would build the list of entities, and then delete all entities in the list.

**Example 4.** An IRDS procedure can be used to simplify the entry of repetitive commands.

Defining the procedure:

```
ADD ENTITY Add-Contains-Rel
  ENTITY-TYPE = IRDS-PROCEDURE
  DESCRIPTIVE-NAME = Add Contains Relationships
  WITH ATTRIBUTES
  DESCRIPTION =
    "Procedure to add relationships from an entity to a list of
    entities. Up to 15 contained entries can be specified in their
    correct relative position."
  DEFINITION =
    " ARG E-
    NAME,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15 ;
    VIEW(IRD,Ex-View)
    DO I = 2 TO ARG() ;
      ADD RELATIONSHIP E-NAME CONTAINS ARG(I)
        WITH REL-POS = I-1 ;
    END ;
    RETURN ; " ;
```

## APPENDIX

Executing the procedure:

```
RUN IRDS-PROCEDURE Add-Contains-Rel
    PASSING file-a, record-a, record-b, record-c, record-d
    RETURN ON ERROR
    SHOW IRDS-COMMANDS ;
```

would generate the following irds-commands and execute each one:

```
ADD RELATIONSHIP file-a CONTAINS record-a WITH REL-POS = 1 ;
    /* output from command */
ADD RELATIONSHIP file-a CONTAINS record-b WITH REL-POS = 2 ;
    /* output from command */
ADD RELATIONSHIP file-a CONTAINS record-c WITH REL-POS = 3 ;
    /* output from command */
ADD RELATIONSHIP file-a CONTAINS record-d WITH REL-POS = 4 ;
    /* output from command */
```

**Example 5.** Similarly, a slightly more complicated IRDS procedure can add the contained entity before adding the relationship.

Defining the procedure:

```
ADD ENTITY Add-Contains-Ent-and-Rel
    ENTITY-TYPE = IRDS-PROCEDURE
    DESCRIPTIVE-NAME = Add Contains Entities and Relationships
    WITH ATTRIBUTES
    DESCRIPTION =
        "Procedure to add a list of entities and the relationships from
        an entity to the list of entities. Entities must be specified
        in their correct relative position."
    DEFINITION =
        " VIEW(IRD,Ex-View)
        /* For each contained entity... */
        DO I = 1 TO ARG()*3 ;
            NAME = I*3 ;
            ETYPE= I*3-1 ;
            DNAME = I*3+1 ;
        /* Add entity with or without descriptive-name,
        depending on whether it was specified. */
        IF ARG(DNAME,E)
            THEN
                ADD ENTITY ARG(NAME) ENTITY-TYPE =
                    ARG(ETYPE)
                DESCRIPTIVE-NAME = ARG(DNAME) ;
            ELSE
                ADD ENTITY ARG(NAME) ENTITY-TYPE =
                    ARG(ETYPE) ;
        /* If error in adding entity: */
        /* (i) do not add relationship */
        /* (ii) set return code from procedure to show it
        only partially executed, and */
        /* (iii) display warning text. */
```

```

CASE WHEN RC = ""ok"" THEN NOP;
      WHEN RC = ""warn"" THEN NOP;
      OTHERWISE RCODE = "partial";
      DISPLAY TEXT
        ""Warning: Relationship not added. ""
        ITERATE;
      END;
/* Add relationship between entity and contained
entity. */
      ADD RELATIONSHIP ARG(1) CONTAINS ARG(NAME)
      WITH REL-POS = I ;
      IF RC = ""error"" THEN RCODE = ""partial"";
/* If error in adding relationship, set return code
from procedure to show it only partially executed.
*/
      END ;
      IF RCODE = ""partial"" THEN RETURN ""partial"";
      RETURN; " ;

```

Executing the procedure:

```

RUN IRDS-PROCEDURE Add-Contains-Ent-and-Rel
      PASSING cust-ad, ELEMENT, cust-name, "Customer Name",
      ELEMENT, cust-street, ,
      ELEMENT, cust-city, "Customer City/State";

```

would be equivalent to the user executing the following commands:

```

ADD ENTITY cust-name ENTITY-TYPE = ELEMENT
      DESCRIPTIVE-NAME = Customer Name ;
ADD RELATIONSHIP cust-ad CONTAINS cust-name
      WITH REL-POS = 1 ;
ADD ENTITY cust-street ENTITY-TYPE = ELEMENT ;
ADD RELATIONSHIP cust-ad CONTAINS cust-street
      WITH REL-POS = 2 ;
ADD ENTITY cust-city ENTITY-TYPE = ELEMENT
      DESCRIPTIVE-NAME = Customer City/State ;
ADD RELATIONSHIP cust-ad CONTAINS cust-city
      WITH REL-POS = 3 ;

```

**Example 6.** A "Help" procedure might be used in conjunction with a procedure to describe the required arguments. For Example 5 above, it might take the following form:

```

ADD ENTITY Help-Add-Contains-Ent-and-Rel
      ENTITY-TYPE = IRDS-PROCEDURE
      DESCRIPTIVE-NAME = Help Add Contains Entities and Relationships
      WITH ATTRIBUTES
      DESCRIPTION =
      "Display required arguments for executing procedure Add-
      Contains-Ent-and-Rel "

```

APPENDIX

```
DEFINITION =  
" DISPLAY TEXT  
  "" 1st argument: Access-name of entity to be updated.  
    2nd - Nth arguments: As many of the following  
      arguments as necessary, in groups of three:  
      (1) contained entity entity-type  
      (2) contained entity access-name  
      (3) contained entity descriptive-name  
          (blank if none).  
    Note: Contained entities must be specified in  
          their correct relative position.  ""  
RETURN; " ;
```

Executing the procedure:

```
RUN IRDS-PROCEDURE Help-Add-Contains-Ent-and-Rel;
```

would result in the following being displayed:

```
1st argument: Access-name of entity to be updated.  
  
2nd - Nth arguments: As many of the following arguments as  
necessary, in groups of three:  
  
      (1) contained entity entity-type  
      (2) contained entity access-name  
      (3) contained entity descriptive-name (blank if none).  
  
Note: Contained entities must be specified in their  
      correct relative position.
```

This is the end of the examples.

**Chapter 6****Module 6 - Application Program Interface****CONTENTS**

<b>SECTION</b>	<b>PAGE</b>
1 Scope, Purpose, and Application . . . . .	6-1
1.1 Scope . . . . .	6-1
1.2 Purpose . . . . .	6-1
1.3 Conformance . . . . .	6-1
1.4 Organization . . . . .	6-2
1.5 Notation . . . . .	6-2
2 References . . . . .	6-2
3 Module Overview . . . . .	6-2
4 Application Program Interface Description . . . . .	6-3
5 Modifications to Module 3 . . . . .	6-4
5.1 New Attribute-Type . . . . .	6-5
5.2 Entity-Type to Attribute-Type Meta-Relationship . . . . .	6-5





## **Chapter 6**

### **Module 6 - Application Program Interface**

#### **1 Scope, Purpose, and Application**

##### **1.1 Scope**

Module 6 of the standard IRDS defines a means by which an application program residing outside of the standard IRDS may access the IRD and IRD Schema. The facilities specified are dependent on the facilities provided in Module 1.

Availability of this Module of the standard IRDS is meaningful only in the event that an implementation of Module 1 of the standard IRDS contains the Command Language Interface.

##### **1.2 Purpose**

This specification of the Application Program Interface defines the functionality of a facility which allows installations and implementors to access the IRD and the IRD Schema from an application program written in a programming language with a "Call" feature. An application may utilize IRDS commands available for IRD Schema maintenance, IRD Schema output, IRD maintenance, IRD output, operations on IRD entity-lists, and certain IRDS utilities.

The specification assumes use of the syntax and semantics of the command language which operates against the IRD and the IRD Schema. The command language syntax is a parameter in the "Call" to the IRDS. Thus, an application programmer uses the same command string as any user of the command language would enter interactively. The results are returned a "record at a time". The records would represent a meta-entity, meta-relationship, entity, or relationship and their respective, associated meta-attributes and attributes.

The need for the Application Program Interface is based on the recognition that with such a facility as part of the IRDS tool set, an installation could provide access the contents of the IRD and the IRD Schema to executing application programs. It is equally recognized that the Application Program Interface does not provide an optimum facility for making the IRDS active in its processing environment. A facility which will make this possible is being defined as a separate standard entitled the IRDS Services Interface, and will be a "low-level" interface oriented towards navigation of the IRD and the IRD Schema.

##### **1.3 Conformance**

The Definition of a Conformant Implementation of Module 6 of a standard IRDS is given in the Definition of a Conformant Implementation of the IRDS (see Section 4 of Requirements for a Conformant Implementation).

## 1.4 Organization

This Module of the standard IRDS consists of the following:

- (1) A description of the Application Program Interface facility.
- (2) Modifications to the IRD Schema.
- (3) Modifications to Module 3 of the standard IRDS, the IRDS Security Module.

The Application Program Interface facility described in this Module has no effect on the Command Language specified in Module 1.

Modifications to Module 3 of the standard IRDS consist of adding one additional attribute-type to the IRD Schema for Module 3, along with a meta-relationship.

Section 3 provides a discussion of the technical basis for this Module of the standard IRDS.

Section 4 provides a full description of the Application Program Interface.

The modifications to Module 3 of the standard IRDS are provided in Section 5.

This Module of the standard IRDS should be reviewed sequentially.

## 1.5 Notation

Where special notation is used, the same notational conventions as identified in Module 1 of the standard IRDS are followed.

The Application Program Interface Description provided in this Module have cross-references to syntax defined both in this Module and in Module 1. Whenever a cross reference number is given as a sequence of numbers alone, the reference is to syntax defined within this Module. References to Module 1 begin with M1. Thus, "M1.5.2" denotes a reference to Subsection 5.2 of Module 1.

## 2 References

None.

## 3 Module Overview

An implementation of the specifications provided by this Module shall result in a programmatic interface to the standard IRDS. That is, an installation and/or an implementor may develop software which could use this interface to access the data in the IRD and the IRD Schema. Hence, this Module could be used by an installation to extend the functionality of Module 1 of the standard IRDS by developing tailored software which can use and update the information found in the IRD Schema and/or the IRD.

The standard IRDS is accessed through the Application Program Interface by an executing program written in a programming language which has a "CALL" feature. In this way, the standard IRDS is treated as an external "subroutine" by the application.

Certain parameters shall be provided through any programming language CALL feature used to access the interface. These parameters include:

- (1) A string of characters which denote a standard IRD Schema or IRD command.
- (2) The designation of a return area for output generated by the specified command.
- (3) A parameter for receiving condition identifiers returned by the standard IRDS.
- (4) Other applicable parameters as defined by the implementor.

This Module shall also consist of a process by which the IRDS user who is accessing the IRDS can be identified. All standard IRDS rules for those Modules of the standard IRDS implemented shall be enforced during the use of this interface.

#### 4 Application Program Interface Description

**Function** To provide a programmatic interface to a standard IRDS.

**Format**

/\* None. This breakdown is explanatory only. \*/

```
application-program-call ::=
    irds-call-statement
    irds-command-parameter
    return-area-parameter
    condition-identifier-return-parameter
    { implementor-defined-parameter }
```

```
irds-call-statement ::=
    [ language-dependent-statement ]
    entry-point
```

```
language-dependent-statement ::=
    /* language-defined-format */
```

```
entry-point ::= IRDS
```

```
irds-command-parameter ::=
    IRD-schema-command M1.5.1
    | IRD-command M1.5.2
    | set-session-defaults-command M1.5.3.2.1
    | session-status-command M1.5.3.2.2
    | exit-IRD-system-command M1.5.3.2.4
```

```
return-area-parameter ::=
    /* implementor-defined-format */

condition-identifier-return-parameter ::=
    /* Error and Warning Conditions for valid IRDS commands.
       See applicable command specifications in Module 1. */

implementor-defined-parameter ::=
    /* implementor-defined-format */
```

**Syntax Rules** None.

### **General Rules**

- (1) Upon identification of the IRDS user, an IRDS session shall be established. The IRDS session shall terminate upon execution of the EXIT command and/or upon another, implementor-defined, action.
- (2) The implementor shall define how many condition identifiers are to be returned in the event of multiple Error/Warning conditions occurring concurrently.
- (3) The implementor shall make session status information available to the user at all times.

### **Access Rules**

The following rule shall apply if Module 3 is installed:

- (1) In order to perform any command execution against the IRD, this interface shall ensure that the IRDS user in effect has the API-PERMISSION attribute equal to YES. See Section 6 of this Module.

### **Error and Warning Conditions**

- (1) Error E06001: Interface not authorized to user. Access Rule (1) of this section has been violated.

## **5 Modifications to Module 3**

Modifications to Module 3 of the standard IRDS, "IRDS Security", are discussed in this section.

This Module requires that one new attribute-type (API-PERMISSION) be added to the IRD Schema as modified by Module 3 of the standard IRDS. This new attribute-type shall then be associated with the IRDS-USER entity-type.

The required changes are specified in the form of an IRD Schema command, although an implementation of the IRDS need not use this approach for these modifications. Likewise, these commands do not specify the movement of the new meta-entity into the CONTROLLED IRD Schema life-cycle-phase, which would be required to support this Module.

Note that the PURPOSE meta-attribute is used as a documentation technique for the attribute-type. In an actual implementation, however, this standard does not require the PURPOSE meta-

attribute. An implementation may use the PURPOSE meta-attribute supplied, or use its own wording. The wording specified for the meta-attribute may be regarded as a suggestion for how the attribute-type required for this Module should be documented in the IRD Schema.

### 5.1 New Attribute-Type

/\* The following attribute-type, in addition to the IRD Schema modifications specified in Module 3, is required to support the Application Program Interface Module. \*/

```

ADD META-ENTITY API-PERMISSION
  META-ENTITY-TYPE = ATTRIBUTE-TYPE
  WITH META-ATTRIBUTES
    ORIGIN = NSX3.13806-1988
    FORMAT = STRING
    MINIMUM-ATTRIBUTE-LENGTH = 1
    MAXIMUM-ATTRIBUTE-LENGTH = 1
    PURPOSE =
      "This attribute-type is associated with the entity-type IRDS-
      USER. Valid values for the attribute-type are 'YES' and 'NO'.
      For a given IRDS-USER entity, a 'YES' indicates that the use of
      the Application Program Interface is permitted."
  ;

```

/\* Attributes of this attribute-type have only two allowable values. The following commands define and implement this validation. \*/

```

ADD META-RELATIONSHIP API-PERMISSION USES VALUE-VALIDATION;
ADD META-RELATIONSHIP API-PERMISSION USES YES-OR-NO-VALUE;

```

### 5.2 Entity-Type to Attribute-Type Meta-Relationship

/\* The following meta-relationship is required. \*/

```

ADD META-RELATIONSHIP
  IRDS-USER CONTAINS API-PERMISSION
  WITH META-ATTRIBUTES
    SINGULAR = YES
    ORIGIN = X3.138-1988 (Ch.6)
  ;

```



**Chapter 7****Module 7 - Entity Lists****CONTENTS**

<b>SECTION</b>	<b>PAGE</b>
1 Scope, Purpose, and Application . . . . .	7-1
1.1 Scope . . . . .	7-1
1.2 Purpose . . . . .	7-1
1.3 Conformance . . . . .	7-1
1.4 Organization . . . . .	7-1
1.5 Notation . . . . .	7-2
1.6 Relationship to Other Modules . . . . .	7-3
2 References . . . . .	7-3
3 Module Overview . . . . .	7-3
4 Basic Language Elements . . . . .	7-4
5 IRDS Entity-List Commands . . . . .	7-4
5.1 Build Entity-List Command . . . . .	7-5
5.2 Entity-List Union Command . . . . .	7-7
5.3 Entity-List Intersection Command . . . . .	7-8
5.4 Entity-List Difference Command . . . . .	7-9
5.5 Entity-List Subtraction Command . . . . .	7-10
5.6 Name Current Entity-list Command . . . . .	7-12
5.7 Output Entity-List Command . . . . .	7-12
5.8 Output Entity-List Names Command . . . . .	7-13
6 Command-Clauses . . . . .	7-14
6.1 Using List Command-Clause . . . . .	7-14
6.2 List Name Command-Clause . . . . .	7-14
7 IRD Entity-List Panel Trees . . . . .	7-15
7.1 Build Entity-List Panel Tree . . . . .	7-16
7.2 Entity-List Union Panel Tree . . . . .	7-17
7.3 Entity-List Intersection Panel Tree . . . . .	7-17
7.4 Entity-List Difference Panel Tree . . . . .	7-18
7.5 Entity-List Subtraction Panel Tree . . . . .	7-19
7.6 Name Current Entity-List Panel Tree . . . . .	7-19
7.7 Output Entity-List Panel Tree . . . . .	7-20
7.8 Output Entity-List Names Panel Tree . . . . .	7-20
8 Modifications to Module 1 . . . . .	7-21
8.1 Modifications to the Command Language . . . . .	7-21
8.1.1 Modifications to the Introduction to Section 5 of Module 1 . . . . .	7-21
8.1.2 Modification to Subsection 5.2, IRD Commands . . . . .	7-21
8.1.3 Modifications to Subsection 5.2.1.3, Delete Entity Command . . . . .	7-22

SECTION	PAGE
8.1.4	Modifications to Subsection 5.2.2, IRD Output Commands . . . . . 7-22
8.1.5	Modification to Subsection 5.2.2.1, General Output Command . . . . . 7-23
8.1.6	Modification to Subsection 5.2.2.2, Impact-of-Change Command . . . . . 7-24
8.1.7	Modification to Subsection 5.2.2.3, Output Syntax Command . . . . . 7-24
8.1.8	Modification to Subsection 6.45, Relationship Selection Command-Clause . 7-25
8.2	Modifications to the Panel Interface . . . . . 7-26
8.2.1	Modification to the Operation of the Panel Interface . . . . . 7-26
8.2.2	Modification to Saving a Panel . . . . . 7-26
8.2.3	Modification to Subsection 8.3.1, State Area . . . . . 7-27
8.2.4	Modification to Subsection 8.4, IRD Panel Trees . . . . . 7-27
8.2.5	Modifications to Subsection 8.4.1.2, Modify Entity Panel Tree . . . . . 7-27
8.2.6	Modifications to Subsection 8.4.1.3, Delete Entity Panel Tree . . . . . 7-27
8.2.7	Modification to Subsection 8.4.1.9, Modify Entity Life-Cycle-Phase Panel Tree . . . . . 7-28
8.2.8	Modification to Subsection 8.4.2.1, General Output Panel Tree . . . . . 7-28
8.2.9	Modification to Subsection 8.4.2.2, Output Impact-of-Change Panel Tree . 7-28
8.2.10	Modification to Subsection 8.4.2.3, Output Syntax Panel Tree . . . . . 7-28
9	Modifications to Module 3 . . . . . 7-29
9.1	Modifications to the Command Language . . . . . 7-29
9.1.1	Modifications to Subsection 7.1, Entities In Command-Clause . . . . . 7-29
9.1.2	Modifications to Subsection 7.10, IRD-Views In Command-Clause . . . . . 7-29
9.1.3	Modifications to Subsection 7.11, For Controllers In Command-Clause . . 7-29
9.1.4	Modifications to Subsection 8.2.2, Module 3 Modifications to IRD Output Functions . . . . . 7-30
9.1.5	Module 3 Extensions Required for the Build Entity-List Command . . . 7-30
9.2	Modifications to Panel Trees . . . . . 7-31
9.2.1	Modification to Subsection 10.1, Add Security Panel Tree . . . . . 7-31
9.2.2	Modification to Subsection 10.2, the Modify Security Panel Tree . . . . . 7-31
9.2.3	Modification to Subsection 10.3, Delete Security Panel Tree . . . . . 7-31
9.2.4	Modification to Subsection 10.4, Add Access Key Panel Tree . . . . . 7-31
9.2.5	Modification to Subsection 10.5, Delete Access Key Panel Tree . . . . . 7-31
10	Cross Reference Indices . . . . . 7-31
10.1	Alphabetic Command Index . . . . . 7-31
10.2	Alphabetic Command-Clause Index . . . . . 7-32
10.3	Error and Warning Condition Index . . . . . 7-32
Appendix	- Collected Syntax . . . . . 7-33



## **Chapter 7**

### **Module 7 - Entity Lists**

#### **1 Scope, Purpose, and Application**

##### **1.1 Scope**

Module 7 of the standard IRDS defines the means by which IRD entity-lists can be defined and manipulated.

##### **1.2 Purpose**

The purpose of this Module of the standard for Information Resource Dictionary Systems (IRDSs) is to provide a mechanism for defining lists of entities, and to provide IRD Commands and Command-Clauses which allow for these entity-lists to be used.

##### **1.3 Conformance**

The Definition of a Conformant Implementation of Module 7 of a standard IRDS is given in the Definition of a Conformant Implementation of the IRDS (see Section 4 of Requirements for a Conformant Implementation).

##### **1.4 Organization**

Module 7 of the standard IRDS consists of the following:

- (1) A set of IRD Commands.
- (2) A set of IRD Command-Clauses.
- (3) A set of Panel Trees.
- (4) Modifications to Module 1.
- (5) Modifications to Module 3.

Modules other than 1 and 3 are not impacted by Module 7.

The Command Language defined in this Module is an upward compatible extension to the Command Language defined in Module 1. Wherever appropriate, the syntax components defined in Module 1 are reused.

This Module does not require any modifications to the IRD Schema Definition or Minimal IRD Schema.

The modifications to Module 1 take three forms:

- (1) Additional command-clauses for specific commands.
- (2) Additional or modified rules for specific commands.
- (3) Additional Panel Trees.

Section 3 provides an overview of where entity-lists fit into the standard IRDS.

Section 4 defines the new elements of the Command Language syntax.

The new Commands associated with this Module are defined in Section 5. The format is the same as found in Module 1 for command specifications.

The Command-Clauses which are unique to this Module are defined in Section 6.

The Panel Trees associated with this Module are defined in Section 7.

The modifications to the Command Language and Panel Trees of Module 1 are given in Section 8.

The modifications to the Security Module (Module 3) are described in Section 9.

Section 10 provides a Cross Reference Indices for Commands, Command-Clauses, and Error and Warning conditions defined in this Module.

This Module of the standard IRDS should be reviewed sequentially.

## 1.5 Notation

This Module uses the same notational conventions as identified in Module 1 of the IRDS standard, except as noted below.

The Commands and Command-Clauses defined in this Module have cross-references to syntax defined both in this Module and in other Modules. Whenever a cross reference number is given as a sequence of numbers alone, the reference is to a Section or Subsection within this Module. References to other Modules of this standard begin with  $Mn$ , where  $n$  denotes the number of the other Module of the IRDS standard.

Thus:

"7.2" denotes a reference to Subsection 7.2 of this Module;

whereas

"M1.4.3" is a reference to Subsection 4.3 of Module 1.

Error and warning conditions have the following format:

- (1) One alphabetic character which defines the message level:
  - (a) E identifies an error.
  - (b) W identifies a warning.
- (2) Two digits identify the Module of the standard IRDS where the message originates.

- (3) Three digits identify the message number within the Module of the standard IRDS and type of condition. (Thus error and warning conditions may have the same last three digits, i.e., E07001 and W07001 are two different conditions.)

## 1.6 Relationship to Other Modules

Additional Modules of the standard IRDS which can be developed and which have dependencies on this Module shall be required to define those dependencies within those Modules. This Module identifies its effects only on Modules 1 through 6.

An implementation which provides the functionality of another Module P which depends upon this Module shall conform to this Module in order to be regarded as in conformance with Module P of the standard IRDS.

## 2 References

None.

## 3 Module Overview

This Module provides a means for specifying a list of entities and manipulating this list as a named object. It should be noted that the term "entity-list" is not to be confused with the BNF terms "entity-name-list" and "meta-entity-name-list". An "entity-list" is a temporary memory store which is created, modified, and accessed via a special set of commands. The terms "entity-name-list" and "meta-entity-name-list" identify the forms for specifying entities and meta-entities directly in a command.

Commands are provided which can produce new entity-lists by set-theoretic operations, i.e., union, intersection, etc., on existing entity-lists. Additionally, entity-lists can be used in place of entity-name lists or selection criteria in certain commands.

An entity list contains zero or more access-names of IRD entities. In the same manner as for commands in Module 1 which process multiple entities, for commands which specify entity-lists, all validation specified by general rules and security rules shall be performed on an entity-by-entity basis for each entity in the list. Additionally,

- (1) If the named entity-list does not exist, an error message shall be produced.
- (2) If the named entity-list exists and is empty, a warning message shall be produced.
- (3) If an error occurs in the processing of any entity within the list, all appropriate error messages shall be produced, and the command shall continue processing with the next entity within the entity-list.

## 4 Basic Language Elements

**Function** To identify those additional terminal and nonterminal symbols which are provided with this Module.

**Format**

```
/* These words are additional irds-words. */  
/* The presentation format is the same as for irds-words. */
```

```
build-word ::= BUILD  
current-word ::= CURRENT  
difference-word ::= DIFFERENCE  
entity-list-word ::= ENTITY-LIST  
intersect-word ::= INTERSECT  
name-word ::= NAME  
names-word ::= NAMES  
subtract-word ::= SUBTRACT  
union-word ::= UNION
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

## 5 IRDS Entity-List Commands

**Function** To create and manipulate entity-lists.

**Format**

```
/* None. This breakdown is explanatory only. */
```

```
IRD-entity-list-command ::=  
    build-entity-list-command           5.1  
    | entity-list-union-command         5.2  
    | entity-list-intersection-command  5.3  
    | entity-list-difference-command    5.4  
    | entity-list-subtraction-command   5.5  
    | name-current-entity-list-command  5.6  
    | output-entity-list-command        5.7  
    | output-entity-list-names-command  5.8
```

**Syntax Rules** None.

**General Rules**

(1) An entity-list is a temporary memory which contains entity-access-names.

- (2) An entity-access-name shall exist in a given entity-list only once.
- (3) As seen by a user, entity-access-names shall be sequenced within a list by assigned-access-name, variation-name, and revision-number. (That is, when an entity-list is input to another IRD command, entities shall be selected from the list in entity-access-name sequence. Also, if the user displays the entity-list via an output-entity-list command, the contents of the list shall be displayed in entity-access-name sequence. The implementor may use any data structuring technique to achieve these results.)
- (4) One entity-list, called the current list, shall always exist. When the IRDS-user enters the IRDS, this list shall be empty.
- (5) When an IRDS-user creates a new current-list, and the existing current-list is not empty, the contents of the existing current-list shall be replaced by that of the new current-list.
- (6) Limitations on the size of an entity-list, and the number of entity-lists which may exist concurrently shall be implementation-dependent.
- (7) All entity-lists for a given IRDS-user shall be deleted when the IRDS-user exits the IRDS. If the implementation permits the user to sign-on under a different IRDS-user name without exiting the IRDS, entity-lists associated with the previously effective IRDS-user shall be deleted.
- (8) An empty, named entity-list can exist.
- (9) For a particular IRDS-user, all entity-list names shall be unique. In the case where different IRDS-users each have an entity-list with the same name, these shall be considered as distinct entity-lists by the IRDS.
- (10) All entity-lists other than the current-list shall be named by the IRDS-user.
- (11) An entity-list which is created by a given effective IRDS-user is visible only to that IRDS-user. That is, if two IRDS-users are active in the IRDS simultaneously, each IRDS-user shall be aware of only those entity-lists which he/she has created.

**Security Rules**     None.

**Actions**         None.

**Error and Warning Conditions**     None.

### 5.1 Build Entity-List Command

**Function**        To create an entity-list.

**Format**

```

build-entity-list-command ::=
    build-word                4
    entity-list-word          4
    entity-selection-criteria-clause  M1.6.33
    [ list-name-clause ]      M1.6.16
    [ using-IRD-views-clause ] M1.6.38
    
```

```
    [ implementor-defined-clauses ]  
    ;  
  
    implementor-defined-clauses ::=  
    /* implementor-defined-format */
```

### Syntax Rules

- (1) The order of command-clauses shall not be significant.

### General Rules

- (1) If using-IRD-views-clause is not specified, then the IRD-View which is in effect shall be used as a default.
- (2) If using-IRD-views-clause is specified, then each IRD-View specified shall exist in the IRD, and be related to the IRDS-USER which is in effect.
- (3) If list-name-clause specifies an existing non-empty entity-list, a warning condition shall be raised. If the command executes, the existing list shall be overwritten.
- (4) If no list-name-clause is specified, the current-list shall be used.

**Security Rules** None.

### Actions

- (1) The command shall be validated according to all Syntax Rules and General Rules. If the command is not valid, all appropriate error and warning messages shall be issued. If the command is valid, then:
  - (a) Entities shall be selected according to selection-criteria-clause and placed in the specified or current-list. If the list existed and was not empty prior to execution of the command, its contents shall be completely replaced by the entity-access-names obtained in the current command execution. If the command does not obtain any entity-access-names, the resulting list shall be empty, even if the resultant entity-list existed and was not empty prior to the execution of this command.
  - (b) The system shall confirm command completion. If the command results in the creation of any empty list, or the replacement of an existing, non-empty list by an empty list, a warning condition shall be raised.

### Error and Warning Conditions

- (1) Error E07111: Invalid IRD-View. General Rule (2) has been violated.
- (2) Warning W07016: Entity-list exists and shall be overwritten. See General Rule (3).
- (3) Warning W07003: Specified entity-list is empty. See Action (1)(b).

NOTE: The implementor shall maintain a count of the number of entity-access-names within each entity-list. The implementor shall provide an information-level message which shall identify the number of entity-access-names within an entity-list when it is created.

## 5.2 Entity-List Union Command

**Function** To create a new entity-list which shall be the set theoretic union of the entity-access-names in two or more entity-lists.

### Format

```

entity-list-union-command ::=
    union-word                                4
    ( existing-entity-list-names )
    [ = new-entity-list-name ]
    ;

existing-entity-list-names ::=
    existing-entity-list-name
    [, existing-entity-list-names

existing-entity-list-name ::=
    entity-list-name
    | null-mark                                M1.4.1

new-entity-list-name ::=
    entity-list-name

entity-list-name ::= irds-name

```

### Syntax Rules

- (1) At least two entity-lists shall be specified within existing-entity-list-names.
- (2) A null-mark within existing-entity-list-names shall identify the current-list.
- (3) If an entity-list-name is duplicated, a warning condition shall be raised.
- (4) If new-entity-list-name is not specified, the current-list shall be assumed.

### General Rules

- (1) Each entity-list-name in existing-entity-list-names shall exist.
- (2) If new-entity-list-name identifies an existing non-empty entity-list, or if the new-entity-list-name is not specified and the current-list is not empty, a warning condition shall exist.

**Security Rules** None.

### Actions

- (1) The command shall be validated. All appropriate error and warning messages shall be issued.
- (2) If the command is valid, the new entity-list shall be created. The resultant entity list shall contain the entity-access-names of entities which were in any of the specified existing entity-lists.

- (3) Completion of command shall be confirmed.

### **Error and Warning Conditions**

- (1) Error E07113: Two entity-lists required. Syntax Rule (1) has been violated.
- (2) Warning W07017: Duplicate entity-list specified. See Syntax Rule (3).
- (3) Error E07016: Specified entity-list does not exist. General Rule (1) has been violated.
- (4) Warning W07018: Resultant entity-list is empty. See General Rule (2).

### **5.3 Entity-List Intersection Command**

**Function** To create a new entity-list whose contents are the set theoretic intersection of two or more existing entity-lists.

#### **Format**

```
entity-list-intersection-command ::=
    intersect-word                               4
    ( existing-entity-list-names )
    [ = new-entity-list-name ]
    ;

existing-entity-list-names ::=
    existing-entity-list-name
    [, existing-entity-list-names ]

existing-entity-list-name ::=
    entity-list-name
    | null-mark                                  M1.4.1

new-entity-list-name ::=
    entity-list-name

entity-list-name ::= irds-name
```

#### **Syntax Rules**

- (1) At least two entity-lists shall be specified within existing-entity-list-names.
- (2) A null-mark within existing-entity-list-names shall identify the current-list.
- (3) If an entity-list-name is specified more than once within existing-entity-list-names, a warning condition shall be raised.
- (4) If new-entity-list-name is not specified, the current-list shall be assumed.

#### **General Rules**

- (1) Each entity-list-name in existing-entity-list-names shall exist.



- (2) If new-entity-list-name identifies an existing non-empty entity-list, or if the new-entity-list-name is not specified and the current-list is not empty, a warning condition shall be raised.

**Security Rules** None.

**Actions**

- (1) The command shall be validated. All appropriate error and warning messages shall be issued.
- (2) If the command is valid, the new entity-list shall be created. The resultant entity-list shall contain the entity-access-names of entities which were in all of the specified existing entity-lists.
- (3) Completion of command shall be confirmed.

**Error and Warning Conditions**

- (1) Error E07113: Two entity-lists required. Syntax Rule (1) has been violated.
- (2) Warning W07017: Duplicate entity-list specified. See Syntax Rule (3).
- (3) Error E07016: Specified entity-list does not exist. General Rule (1) has been violated.
- (4) Warning W07018: Resultant entity-list is empty. See General Rule (2).

**5.4 Entity-List Difference Command**

**Function** To create a new entity-list whose contents shall be the symmetric difference of two other entity-lists. The symmetric difference of two sets is defined as the complement of the intersection of two sets. This is formally stated as follows:

Given two sets  $A = \{a\}$  and  $B = \{b\}$ , the symmetric difference  $D$  is defined as follows:

$$D = \{ d \text{ such that } (d \text{ in the union of } A \text{ and } B) \text{ and } (d \text{ not in the intersection of } A \text{ and } B) \}$$

**Format**

```
entity-list-difference-command ::=
    difference-word                               4
    ( entity-list-1-name , entity-list-2-name )
    [ = new-entity-list-name ]
    ;
```

```
entity-list-1-name ::=
    entity-list-name
    | null-mark                                   M1.4.1
```

```
entity-list-2-name ::=
    entity-list-name
    | null-mark
```

M1.4.1

```
new-entity-list-name ::=
    entity-list-name
```

```
entity-list-name ::= irds-name
```

### Syntax Rules

- (1) A null-mark within existing-entity-list-names shall identify the current-list.
- (2) If an entity-list-name is duplicated, a warning condition shall be raised.
- (3) If new-entity-list-name is not specified, the current-list shall be assumed.

### General Rules

- (1) Entity-list-1-name and entity-list-2-name shall exist.
- (2) If new-entity-list-name identifies an existing non-empty entity-list, or if the new-entity-list-name is not specified and the current-list is not empty, a warning condition shall be raised.

**Security Rules**     None.

### Actions

- (1) The command shall be validated. All appropriate error and warning messages shall be issued.
- (2) If the command is valid, the new entity-list shall be created. The resultant entity-list shall contain the entity-access-names of entities which are in the symmetric difference of the specified existing entity-lists.
- (3) Completion of command shall be confirmed.

### Error and Warning Conditions

- (1) Error E07113: Two entity-lists required. Syntax Rule (1) has been violated.
- (2) Warning W07017: Duplicate entity-list specified. See Syntax Rule (3).
- (3) Error E07016: Specified entity-list does not exist. General Rule (1) has been violated.
- (4) Warning W07018: Resultant entity-list is empty. See General Rule (2).

## 5.5 Entity-List Subtraction Command

**Function**     To create a new entity list from an existing entity-list *A*, by removing those entity-access-names which *A* has in common with another specified entity list *B*. This is formally specified as follows:

Given two sets  $A = \{a\}$  and  $B = \{b\}$ , the set  $A-B$  is defined as follows:

$$A-B = \{ s \text{ such that } (s \text{ in } A) \text{ and } (s \text{ not in the intersection of } A \text{ and } B) \}$$

### Format

```
entity-list-subtraction-command ::=
    subtract-word                                4
    ( entity-list-1-name , entity-list-2-name )
    [ = new-entity-list-name ]
    ;

entity-list-1-name ::=
    entity-list-name
    | null-mark                                M1.4.1

entity-list-2-name ::=
    entity-list-name
    | null-mark                                M1.4.1

new-entity-list-name ::=
    entity-list-name

entity-list-name ::= irds-name
```

### Syntax Rules

- (1) A null-mark within existing-entity-list-names shall identify the current-list.
- (2) If an entity-list-name is duplicated, a warning condition shall be raised.
- (3) If new-entity-list-name is not specified, the current-list shall be assumed.

### General Rules

- (1) Entity-list-1-name and entity-list-2-name shall exist.
- (2) If new-entity-list-name identifies an existing entity-list, or if the new-entity-list-name is not specified and the current-list is not empty, a warning condition shall be raised.

**Security Rules**    None.

### Actions

- (1) The command shall be validated. All appropriate error and warning messages shall be issued.
- (2) If the command is valid, the new entity-list shall be created. The resultant entity list shall contain the access-names of entities which are in entity-list-1-name but not common to entity-list-1-name and entity-list-2-name.
- (3) Completion of command shall be confirmed.

### **Error and Warning Conditions**

- (1) Error E07113: Two entity-lists required. Syntax Rule (1) has been violated.
- (2) Warning W07017: Duplicate entity-list specified. See Syntax Rule (3).
- (3) Error E07016: Specified entity-list does not exist. General Rule (1) has been violated.
- (4) Warning W07018: Resultant entity-list is empty. See General Rule (2).

### **5.6 Name Current Entity-list Command**

**Function** To assign a user-specified name to the current-list.

#### **Format**

```
name-current-entity-list-command ::=
    name-word           4
    current-word        4
    entity-list-word    4
    entity-list-name
    ;

entity-list-name ::= irds-name
```

**Syntax Rules** None.

#### **General Rules**

- (1) Entity-list-name shall not specify the name of an existing entity-list.

**Security Rules** None.

#### **Actions**

- (1) If the command has any errors, all appropriate error messages shall be produced.
- (2) If the command passes validation, the current-list shall be given the specified name. A new, empty current-list shall be created.
- (3) Confirmation of completion of command processing shall be given.

### **Error and Warning Conditions**

- (1) Error E07114: Specified entity-list name already exists. General Rule (1) has been violated.

### **5.7 Output Entity-List Command**

**Function** To display the contents of a specified entity-list.

**Format**

```

output-entity-list-command ::=
    output-word           M1.4.3
    entity-list-word      4
    [ list-name-clause ] M1.6.16
    [ show-title-clause ] M1.6.65
    [ route-clause ]     M1.6.64
    ;

```

**Syntax Rules**

- (1) If list-name-clause is not specified, the current-list shall be assumed.

**General Rules**

- (1) If list-name-clause is specified, the specified entity-list-name shall identify an existing entity-list.

**Security Rules** None.

**Actions**

- (1) If the command passes validation, the contents of the entity-list shall be formatted and routed to the specified or default destination. Otherwise, all appropriate error messages shall be produced. Completion of command processing shall be confirmed.

The output shall show all entity-access-names in the entity-list in entity-access-name sequence. It shall also provide the count of the number of entity-access-names within the entity-list. The name of the effective IRDS-user, and a date and time stamp of command execution shall also be displayed.

**Error and Warning Conditions**

- (1) Error E07016: Specified entity-list does not exist. General Rule (1) has been violated.

**5.8 Output Entity-List Names Command**

**Function** To display the names of all entity lists which currently exist for an IRDS-user.

**Format**

```

output-entity-list-names-command ::=
    output-word           M1.4.3
    entity-list-word      4
    names-word            4
    [ show-title-clause ] M1.6.65
    [ route-clause ]     M1.6.64
    ;

```

**Syntax Rules** None.

**General Rules** None.

**Security Rules** None.

**Actions**

- (1) A listing of all currently existing entity-lists for the effective IRDS-user is formatted and routed to the specified destination. The listing shall display the names of the entity-lists in name sequence. For each entity-list name, the number of entity-access-names within the list shall also be shown. The name of the effective IRDS-user and a date and time stamp of command execution shall also be displayed.

**Error and Warning Conditions** None.

## 6 Command-Clauses

### 6.1 Using List Command-Clause

**Function** To identify an entity-list to be used as input to a command.

**Format**

```
using-list-clause ::=
    [ using-word ]           4
    entity-list-word        4
    [ = entity-list-name ]

entity-list-name ::= irds-name
```

**Syntax Rules** None.

**General Rules**

- (1) If list-name is not specified, the current-list shall be assumed.
- (2) List-name shall identify an existing entity-list.
- (3) If the list identified by list-name is empty, a warning condition shall be issued.

**Error and Warning Conditions**

- (1) Error E07016: Specified entity-list does not exist. General Rule (2) of this command-clause has been violated.
- (2) Warning W07003: Specified entity-list is empty. See General Rule (3) of this command-clause.

### 6.2 List Name Command-Clause

**Function** To assign a name to an entity-list which is to be created.

**Format**

```
list-name-clause ::=
    list-name-word                               4
    =
    entity-list-name

entity-list-name ::= irds-name
```

**Syntax Rules** None.

**General Rules** None.

**Error and Warning Conditions** None.

**7 IRD Entity-List Panel Trees**

**Function** To identify all panel trees which create and manipulate entity-lists, and the rules which apply to all such panel trees.

**Format**

/\* None. This breakdown is explanatory only. \*/

```
IRD-entity-list-panel-tree ::=
    build-entity-list-panel-tree                 7.1
    | entity-list-union-panel-tree               7.2
    | entity-list-intersection-panel-tree        7.3
    | entity-list-difference-panel-tree          7.4
    | entity-list-subtraction-panel-tree         7.5
    | name-current-entity-list-panel-tree        7.6
    | output-entity-list-panel-tree              7.7
    | output-entity-list-names-panel-tree        7.8
```

**Syntax Rules**

- (1) The Syntax Rules which shall exist for user-supplied information in the Data Area shall include those which exist in the specifications for the equivalent functions in the Command Language Interface, except that the Syntax Rules which relate to command-clause ordering within a particular command specification shall not apply. Refer to the IRD Entity-List Commands in the introduction to Section 5 for a description of these rules.

**General Rules**

- (1) The General Rules which shall exist for these panel trees shall include those which exist for the equivalent command, with all of its associated command-clauses, in the Command Language Interface. Refer to the introduction to Section 5 for a description of these rules.

**Security Rules** None.

### **Actions**

- (1) The Actions which shall be performed by this panel tree shall be equivalent to those which are performed by the equivalent command in the Command Language Interface. Refer to the introduction to Section 5 for a description of these actions.

### **Error and Warning Conditions**

- (1) Error and Warning Conditions which can appear in the Message Area of one or more of the panels in the panel trees described in the following subsections are the same as those which exist for the equivalent IRD maintenance command found in the introduction to Section 5.

## **7.1 Build Entity-List Panel Tree**

**Function** This panel tree shall be used to create a new entity-list.

### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the new entity-list to be created. In the case where no entity-list name is entered, the current-list shall be used.
  - (b) The entity selection criteria to be used in specifying the entities to be included in the entity-list. This entity selection criteria shall include the functionality provided in the entity-selection-criteria-clause as specified in Subsection 6.33, and all of its associated subclauses.
  - (c) The IRD-Views to which the entity selection shall be limited. If no IRD-Views are specified, then the IRD-View which is in effect shall be used as a default.

### **IRD Schema Area Contents**

- (1) The IRD Schema Area of one or more panels in this panel tree shall provide the following information to the user:
  - (a) Rules for entity naming.
  - (b) A list of valid entity-types available to the user.
  - (c) A list of valid IRD-partitions.
  - (d) A list of valid quality-indicators.
  - (e) A list of valid variation-names for particular entity-types.
  - (f) A list of valid relationship-types available to the user.
  - (g) Rules for applicable attributes and attribute-groups, including values or ranges for an attribute-type-validation-data descriptor.



**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Create the specified entity-list, in which case the selected entities are assigned to the specified entity-list.
  - (b) Display validation rules, in which case the desired validation information shall appear in the IRD Schema Area.
  - (c) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

**7.2 Entity-List Union Panel Tree**

**Function** This panel tree shall be used to create a new entity-list which is the set theoretic union of the entity-access-names in two or more entity-lists.

**Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the new entity-list to be created. In the case where no entity-list name is entered, the current-list shall be used.
  - (b) The names of two or more existing entity-lists on which the union operation shall be performed. The current-list designation can be used for one of the existing entity-lists.

**IRD Schema Area Contents** None.

**Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Perform the entity-list union on the specified entity-lists, in which case the resulting entities are assigned to the new entity-list specified.
  - (b) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

**7.3 Entity-List Intersection Panel Tree**

**Function** This panel tree shall be used to create a new entity-list which is the set theoretic intersection of the entity-access-names in two or more entity-lists.

### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the new entity-list to be created. In the case where no entity-list name is entered, the current-list shall be used.
  - (b) The names of two or more existing entity-lists on which the intersection operation shall be performed. The current-list designation can be used for one of the existing entity-lists.

**IRD Schema Area Contents** None.

### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Perform the entity-list intersection on the specified entity-lists, in which case the resulting entities are assigned to the new entity-list specified.
  - (b) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

## 7.4 Entity-List Difference Panel Tree

**Function** This panel tree shall be used to create a new entity-list which is the set theoretic symmetric difference of the entity-access-names in two entity-lists.

### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the new entity-list to be created. In the case where no entity-list name is entered, the current-list shall be used.
  - (b) The names of two existing entity-lists on which the symmetric difference operation shall be performed. The current-list designation can be used for one of the existing entity-lists.

**IRD Schema Area Contents** None.

### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Perform the entity-list symmetric difference on the specified entity-lists, in which case the resulting entities are assigned to the new entity-list specified.

- (b) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

### **7.5 Entity-List Subtraction Panel Tree**

**Function** This panel tree shall be used to create a new entity-list which is the set theoretic subtraction of the entity-access-names in two entity-lists.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the new entity-list to be created. In the case where no entity-list name is entered, the current-list shall be used.
  - (b) The names of two existing entity-lists on which the subtraction operation shall be performed. The current-list designation can be used for one of the existing entity-lists.

**IRD Schema Area Contents** None.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Perform the entity-list subtraction on the specified entity-lists, in which case the resulting entities are assigned to the new entity-list specified.
  - (b) Refresh the Data Area, in which case all user-supplied data shall be erased from the Data Area.

### **7.6 Name Current Entity-List Panel Tree**

**Function** This panel tree shall be used to assign a user-specified name to the current entity-list.

#### **Data Area Contents**

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the name to be assigned to the current entity-list.

**IRD Schema Area Contents** None.

#### **Action Area Contents**

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display an option to assign the specified entity-list name to the current entity-list.

### 7.7 Output Entity-List Panel Tree

**Function** This panel tree shall be used to display the contents of a specified entity-list.

#### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) The name of the entity-list whose contents shall be displayed. If no entity-list name is specified, the current entity-list shall be assumed.
  - (b) A title to be displayed on the output of the entity-list contents.
  - (c) The destination to which the output shall be routed. The entry of this information shall be optional. If not entered, an implementation-specified default destination shall be used.

**IRD Schema Area Contents** None.

#### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display an option to display the contents of the specified entity-list, in which case the contents of the specified entity-list shall be routed to the selected destination.

### 7.8 Output Entity-List Names Panel Tree

**Function** This panel tree shall be used to display the names of all entity-lists which currently exist for the user.

#### Data Area Contents

- (1) The Data Area of one or more panels in this panel tree shall request and accept input of the following information:
  - (a) A title to be displayed along with the entity-list names output.
  - (b) The destination to which the output shall be routed. The entry of this information shall be optional. If not entered, an implementation-specified default destination shall be used.

**IRD Schema Area Contents** None.

#### Action Area Contents

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display an option to display the entity-list names, in which case the names of all entity-lists which currently exist for the user are routed to the selected destination.

## 8 Modifications to Module 1

### 8.1 Modifications to the Command Language

#### 8.1.1 Modifications to the Introduction to Section 5 of Module 1

General Rule (4) is replaced by:

- (4) Multiple IRD Schema or IRD objects can be specified directly within the command. Additionally, IRD commands support two techniques for indirectly specifying IRD objects:
  - (a) A selection criteria command-clause.
  - (b) Identification of an entity-list.

/\* Note, the term "entity-list" is not to be confused with the BNF terms "entity-name-list" and "meta-entity-name-list". An "entity-list" is a temporary memory store which is created, modified, and accessed via a special set of commands. The terms "entity-name-list" and "meta-entity-name-list" identify the forms for specifying entities and meta-entities directly in a command. \*/

General Rules (+1) and (+2) are added as follows:

- (+1) An entity-list is a named temporary memory which contains zero or more access-names of IRD entities. For commands which specify entity-lists, all validation specified by general rules and security rules shall be performed on an entity-by-entity basis for each entity in the list.
  - (a) If the named entity-list does not exist, an error message shall be produced.
  - (b) If the named entity-list exists and is empty, a warning message shall be produced.
  - (c) If an error occurs in the processing of any entity within the list, all appropriate error messages shall be produced, and the command shall continue processing with the next entity within the entity-list.
- (+2) The term user means the individual who requests the IRDS to perform one of the functions specified by the commands specified within this section. Each user is presumed to have a corresponding entity of type IRDS-USER defined in the IRD. This entity is called the effective IRDS-user.

#### 8.1.2 Modification to Subsection 5.2, IRD Commands

Format is replaced by:

/\* None. This breakdown is explanatory only. \*/

```

IRD-command ::=
    IRD-maintenance-command           M1.5.2.
    | IRD-output-command                M1.5.2.2
    | IRD-entity-list-command           5
    
```

### 8.1.3 Modifications to Subsection 5.2.1.3, Delete Entity Command

Format is replaced by:

```

delete-entity-command ::=
    delete-word           M1.4.3
    entity-word           M1.4.3
    delete-selection-option
    [ with-relationships-clause ] M1.6.32
    ;

delete-selection-option ::=
    entity-name-list
    | using-list-clause6.1
    | entity-selection-criteria-clause M1.6.33

entity-name-list ::=
    entity-access-name M1.4.4
    [ , entity-name-list ]
    
```

**General Rule (1)** is replaced by:

- (1) Each entity-access-name specified directly in the command or stored in an entity-list shall identify an entity which exists in the IRD.

The following **Warning Condition** is added:

- (+1) Warning W07005: List specifies nonexistent entities.

### 8.1.4 Modifications to Subsection 5.2.2, IRD Output Commands

**General Rule (2)** is replaced by:

- (2) Selection-option is always defined as follows:

```

selection-option ::=
    entity-selection-criteria-clause
    | using-list-clause
    
```

**General Rule (+1)** is added:

- (+1) If using-list-clause is specified:
  - (a) The identified entity-list shall exist.
  - (b) If the identified or current-list is empty, a warning condition shall be raised.
  - (c) If entities whose names are within the entity-list are of entity-types which are not consistent with the show-clauses or predefined display identified in this command, those entities shall be bypassed.

The following **Error and Warning Conditions** are added:

- (+1) Error E07016: Specified entity-list does not exist. General Rule (+1)(a) has been violated.
- (+2) Warning W07003: Specified entity-list is empty. See General Rule (+1)(b).

### 8.1.5 Modification to Subsection 5.2.2.1, General Output Command

Format is replaced by:

```

general-output-command ::=
    output-word                M1.4.3
    IRD-word                   M1.4.3
    [ using-IRD-views-clause ] M1.6.38
    selection-option
    [ sort-clause ]           M1.6.62
    show-options
    [ route-clause ]         M1.6.64
    [ implementor-defined-clauses ]
    ;

selection-option ::=
    entity-selection-criteria-clause M1.6.33
    | using-list-clause              6.1

show-options ::=
    show-predefined-display-clause M1.6.63
    | show-clause-list

show-clause-list ::=
    [ show-title-clause ]           M1.6.65
    unqualified-show-clause-list
    [ qualified-show-clause-list

unqualified-show-clause-list ::=
    general-output-show-clause-option
    [ unqualified-show-clause-list ]

qualified-show-clause-list ::=
    qualified-show-clause           M1.6.67
    [ qualified-show-clause-list ]

general-output-show-clause-option ::=
    show-all-clause                M1.6.74
    | show-entity-type-clause       M1.6.71
    | show-IRD-life-cycle-phase-clause M1.6.68
    | show-entity-access-name-clause M1.6.69
    | show-entity-descriptive-name-clause M1.6.70
    | show-relationships-clause     M1.6.72
    | show-attributes-clause        M1.6.73
    | show-counts-clause            M1.6.82
    | show-quality-indicator-clause M1.6.83

```

```

implementor-defined-clauses ::=
    /* implementor-defined-format */

```

### 8.1.6 Modification to Subsection 5.2.2.2, Impact-of-Change Command

Format is replaced by:

```

output-impact-of-change-command ::=
    output-word M1.4.3
    [ impact-option ]
    impact-word M1.4.3
    [ using-IRD-views-clause ] M1.6.38
    selection-option
    [ sort-clause ] M1.6.62
    [ impact-show-options ]
    [ route-clause ] M1.6.64
    [ implementor-defined-clauses ]
    ;

impact-option ::=
    cumulative-word M1.4.3
    | individual-word M1.4.3

selection-option ::=
    entity-selection-criteria-clause M1.6.33
    | using-list-clause 6.1

impact-show-options ::=
    [ show-title-clause ] M1.6.65
    [ entity-type-show-restriction-clause ] M1.6.66
    { impacted-entity-show-clause }
    { qualified-impacted-entity-show-clause }

impacted-entity-show-clause ::=
    show-IRD-life-cycle-phase-clause M1.6.68
    show-quality-indicator-clause M1.6.83
    show-attributes-clause M1.6.73
    show-entity-descriptive-name-clause M1.6.70

qualified-impacted-entity-show-clause ::=
    entity-type-qualification-clause M1.6.1
    impacted-entity-show-clause-list

impacted-entity-show-clause-list ::=
    impacted-entity-show-clause
    [ impacted-entity-show-clause-list ]

```

### 8.1.7 Modification to Subsection 5.2.2.3, Output Syntax Command

Format is replaced by:

```

output-syntax-command ::
    output-word M1.4.3
    syntax-word M1.4.3

```



[ using-IRD-views-clause ]	M1.6.38
selection-option	
[ sort-clause ]	M1.6.62
[ syntax-show-options ]	
[ route-clause ]	M1.6.64
[ implementor-defined-clauses ]	
;	
selection-option ::=	
entity-selection-criteria-clause	M1.6.33
using-list-clause	6.1
syntax-show-options ::=	
[ show-title-clause ]	M1.6.65
[ show-IRD-life-cycle-phase-clause ]	M1.6.68
[ show-relationships-clause ]	M1.6.72
[ show-relationship-syntax-clause ]	M1.6.75

### 8.1.8 Modification to Subsection 6.45, Relationship Selection Command-Clause

Format is replaced by:

relationship-selection-clause ::=	
select-word	M1.4.3
[ all-word	M1.4.3
[ forward-or-inverse-word ]	
]	
relationships-word	M1.4.3
for-word	M1.4.3
specified-entities	
[ where-word	M1.4.3
relationship-restriction-expression	
]	
forward-or-inverse-word ::=	
forward-word	M1.4.3
inverse-word	M1.4.3
specified-entities ::=	
using-list-clause	6.1
name-scan-pattern-list	
name-scan-pattern-list ::=	
name-pattern	
[ , name-scan-pattern-list ]	
name-pattern ::=	
entity-access-name	M1.4.4
name-scan-pattern	M1.4.4
relationship-restriction-expression ::=	
relationship-restriction-clause	
( relationship-restriction-expression )	
relationship-restriction-expression	

boolean-operator M1.4.3  
 relationship-restriction-expression

relationship-restriction-clause ::=  
     relationship-type-restriction-clause M1.6.50  
     | attribute-restriction-clause M1.6.36  
     | attribute-group-restriction-clause M1.6.37  
     | order-restriction-clause M1.6.26  
     | text-attribute-substring-restriction-clause M1.6.40  
     | irds-function-restriction-clause M1.6.77

**General Rule (+1)** is added as follows:

- (+1) If a using-list-clause is specified, the specified list shall exist. If it exists but is empty, a warning condition shall exist.

The following **Error and Warning Conditions** are added:

- (+1) Error E07016: Specified entity-list does not exist. General Rule (+1) for this command-clause has been violated.
- (+2) Warning W07003: Specified entity-list is empty. See General Rule (+1) for this command-clause.

## 8.2 Modifications to the Panel Interface

### 8.2.1 Modification to the Operation of the Panel Interface

The last paragraph reads as follows:

For operations involving the deletion of entities, relationships, meta-entities or meta-relationships, the user shall have the option of performing the deletion either with or without confirmation. When confirmation is requested, a full display of the entity/meta-entity or the relationship/meta-relationship (including all of the attributes/meta-attributes) shall be provided in the Data Area and the user shall be given the opportunity to proceed with or bypass the deletion. When confirmation is not requested, the deletion shall be performed with no such display in the Data Area. This shall also occur in the event that a user has specified the name of an entity-list in the Delete Entity panel tree. If confirmation is not requested, all entities in the specified entity-list shall be deleted. If confirmation is requested, the system shall loop through each entity in the entity-list, allowing the user to confirm deletion for each entity.

### 8.2.2 Modification to Saving a Panel

The following comment is added:

/\* An example of this feature might be the case where a user is creating an entity-list. If the user requests the option to save a panel when logging off, the system shall save the entity-list panel in which the user was operating and all relevant panels associated with the creation of the entity-list. At the beginning of the next session, the user can request to be placed at the panel at which the user was working when the previous session ended. At this point the original panel and all relevant panels shall be restored to their former state, and the user can continue with the process of creating an entity-list. \*/

**8.2.3 Modification to Subsection 8.3.1, State Area**

The following contents is added to the **State Area**:

- (+1) The name of the entity-list, if any, which is being created or used by the current IRD panel.

**8.2.4 Modification to Subsection 8.4, IRD Panel Trees**

Format is replaced by:

/\* None. This breakdown is explanatory only. \*/

```

panel-tree ::=
    IRD-maintenance-panel-trees           M1.8.4.1
    | IRD-output-panel-trees              M1.8.4.2
    | IRD-entity-list-panel-trees         7
    
```

**8.2.5 Modifications to Subsection 8.4.1.2, Modify Entity Panel Tree**

The **Data Area** of one or more panels in this panel tree:

- (+1) Shall also request input of the following:

The name of an entity-list which contains entity-access-names of existing entities to be sequentially modified. Note that the use of entity-lists for this purpose shall be unique to the Panel Interface.

- (+2) Shall also display the following option to the user:

Loop, in which case the next entity in the specified entity-list shall be displayed for modification.

**8.2.6 Modifications to Subsection 8.4.1.3, Delete Entity Panel Tree**

The **Data Area** of one or more panels in this panel tree shall also request input of the following information:

The name of an entity-list which contains entity-access-names of existing entities to be deleted. Entry of an entity-list name shall be optional. If specified, the entities in the entity-list shall be deleted with or without user confirmation, depending on the option selected in the Action Area of this panel tree.

The **Action Area Contents** is replaced by the following:

- (1) In addition to those global options specified in Subsection 8.3.4 of Module 1, the Action Area of one or more panels in this panel tree shall display the following options to the user:
  - (a) Delete with Confirmation, in which case the entity-access-name, the entity-descriptive-name, the IRD-partition, the quality-indicator, and the attributes and attribute-groups associated with the specified entity are displayed in the Data Area and the user shall be given the opportunity to proceed with or bypass the entity deletion. In the event that the user has specified the name of an entity-list and

wishes to perform delete confirmation for each entity in the list, the loop option shall be used.

- (b) Delete, in which case the entity with the specified entity-access-name or all of the entities in the specified entity-list are deleted from the IRD without confirmation.
- (c) Loop, in which case the entity-access-name, the entity-descriptive-name, the IRD-partition, the quality-indicator, and the attributes and attribute-groups associated with the next entity in the specified entity-list shall be displayed in the Data Area for possible deletion. The user shall be given the opportunity to proceed with or bypass the deletion of each entity in the entity-list.

#### **8.2.7 Modification to Subsection 8.4.1.9, Modify Entity Life-Cycle-Phase Panel Tree**

The **Data Area** of one or more panels in this panel tree shall also request input of the following information:

The name of an entity-list which contains entity-access-names of existing entities to be sequentially transferred from one IRD life-cycle-phase to another.

The **Action Area** of one or more panels in this panel tree shall display the following option to the user:

Loop, in which case the next entity in the specified entity-list shall be displayed for possible transfer from one IRD life-cycle-phase to another.

#### **8.2.8 Modification to Subsection 8.4.2.1, General Output Panel Tree**

In the **Data Area Contents**, (1)(b) is replaced by:

- (b) The selection criteria for the entities to be included in the output. This can include the name of an existing entity-list.

#### **8.2.9 Modification to Subsection 8.4.2.2, Output Impact-of-Change Panel Tree**

In the **Data Area Contents**, (1)(c) is replaced by:

- (c) The selection criteria for the entities whose impact-of-change shall be analyzed. This can include the name of an existing entity-list.

#### **8.2.10 Modification to Subsection 8.4.2.3, Output Syntax Panel Tree**

In the **Data Area Contents**, (1)(b) is replaced by:

- (b) The selection criteria for the entities to be output by this panel tree. This can include the name of an existing entity-list.

#### **8.2.11 Modification to Subsection 8.5.2, Export IRD Panel Tree \*\*EX/IM\*\***

Clause (1)(b) in the **Data Area Contents** is replaced by:

- (b) The specification of the entities which are to be extracted from the current IRD. This information can be in the form of an entity-list, or an indication that only the

IRD Schema shall be extracted. In the latter case, no IRD-export-file shall be produced.

## 9 Modifications to Module 3

### 9.1 Modifications to the Command Language

#### 9.1.1 Modifications to Subsection 7.1, Entities In Command-Clause

Format is replaced by:

```

entities-in-clause ::=
    entities-word           M1.4.3
    in-word                 M1.4.3
    entity-selection-set-spec

entity-selection-set-spec ::=
    [ using-IRD-views-clause ] M1.6.38
    entity-selection-criteria-clause M1.6.33
    | entity-selection-criteria-clause M1.6.33
    [ using-IRD-views-clause ] M1.6.38
    | using-list-clause M1.6.13
    
```

#### 9.1.2 Modifications to Subsection 7.10, IRD-Views In Command-Clause

Format is replaced by:

```

IRD-views-in-clause ::=
    IRD-views-word           M1.4.3
    in-word                 M1.4.3
    entity-selection-set-spec

entity-selection-set-spec ::=
    [ using-IRD-views-clause ] M1.6.38
    entity-selection-criteria-clause M1.6.33
    | entity-selection-criteria-clause M1.6.33
    [ using-IRD-views-clause ] M1.6.38
    | using-list-clause 6.1
    
```

**General Rule (+1)** is added as follows:

(+1) Each entity in the entity-list specified by using-list-clause shall exist in the IRD.

#### 9.1.3 Modifications to Subsection 7.11, For Controllers In Command-Clause

Format is replaced by:

```

for-controllers-in-clause ::=
    for-word           M1.4.3
    controllers-word M3.5.2
    
```

in-word M1.4.3  
 entity-selection-set-spec

```
entity-selection-set-spec ::=
    [ using-IRD-views-clause ] M1.6.38
    entity-selection-criteria-clause M1.6.33
    | entity-selection-criteria-clause M1.6.33
    [ using-IRD-views-clause ] M1.6.38
    | using-list-clause 6.1
```

**General Rule (+1)** is added as follows:

(+1) Each entity in the entity-list specified by using-list-clause shall exist in the IRD.

**9.1.4 Modifications to Subsection 8.2.2, Module 3 Modifications to IRD Output Functions**

**General Rule (+3)** now reads as follows:

(+3) If an entity satisfies the specified selection criteria, but:

- (a) It is of an entity-type which is not visible within the IRD-views used by this command, and
- (b) It is not explicitly specified within the selection criteria,

then it shall be bypassed. Entities identified within an entity-list but not visible within the IRD-views used by a command shall also be bypassed.

**9.1.5 Module 3 Extensions Required for the Build Entity-List Command**

The following **Security Rules** are required:

- (+1) Only entities which are visible to an IRDS-user can be saved in an entity-list. The set of entities which shall be visible to an IRDS-user in a build-entity-list command are those which are visible by the IRD-views specified in the using-IRD-views-clause. If no using-IRD-views-clause is specified, then only those entities visible in the effective IRD-view can be saved in an entity-list. Any entity which meets the selection-criteria but is not visible in the IRD-views used by the command shall be ignored.
- (+2) A selection-criteria shall not specify any entity-type or relationship-type (in the CONTROLLED IRD Schema life-cycle-phase) for which READ-PERMISSION is not authorized.
- (+3) The selection-criteria "WHERE RELATIONSHIPS EXIST" and "WHERE NO RELATIONSHIPS EXIST" shall be evaluated without regard to whether or not the relationships in which the tested entity participates are visible via the specified or defaulted IRD-views.
- (+4) The relationships which implement entity-level security shall not be used in evaluating the selection criteria "WHERE RELATIONSHIPS EXIST" and "WHERE NO RELATIONSHIPS EXIST".

The following **Error Condition** exists:

(+1) Error E07014: IRD Schema descriptor not visible to user. See Security Rule (+2).

**9.2 Modifications to Panel Trees**

**9.2.1 Modification to Subsection 10.1, Add Security Panel Tree**

The **Data Area Contents** contains the additional clauses:

- (1)(a)(i+) The name of an existing entity-list.
- (1)(b)(i+) The name of an existing entity-list.

**9.2.2 Modification to Subsection 10.2, the Modify Security Panel Tree**

The **Data Area Contents** contains the additional clause:

- (1)(a)(i+) The name of an existing entity-list.

**9.2.3 Modification to Subsection 10.3, Delete Security Panel Tree**

The **Data Area Contents** contains the additional clauses:

- (1)(a)(i+) The name of an existing entity-list.
- (1)(b)(i+) The name of an existing entity-list.

**9.2.4 Modification to Subsection 10.4, Add Access Key Panel Tree**

The **Data Area Contents** contains the additional clauses:

- (1)(b)(i+) The name of an existing entity-list.
- (1)(c)(i+) The name of an existing entity-list.

**9.2.5 Modification to Subsection 10.5, Delete Access Key Panel Tree**

The **Data Area Contents** contains the additional clauses:

- (1)(a)(i+) The name of an existing entity-list.
- (1)(b)(i+) The name of an existing entity-list.

**10 Cross Reference Indices**

**10.1 Alphabetic Command Index**

<u>Command</u>	<u>Subsection</u>
build-entity-list-command	5.1
entity-list-difference-command	5.3
entity-list-intersection-command	5.4
entity-list-subtraction-command	5.5
entity-list-union-command	5.2

name-current-entity-list-command	5.6
output-entity-list-command	5.6
output-entity-list-names-command	5.8

**10.2 Alphabetic Command-Clause Index**

<u>Command-Clause</u>	<u>Subsection</u>
list-name-command-clause	6.2
using-list-command-clause	6.1

**10.3 Error and Warning Condition Index**

E07014:	IRD Schema descriptor not visible to user.						
	9.1.5						
E07016:	Specified entity-list does not exist.						
	5.3	5.4	5.5	5.7	6.1	8.1.4	8.1.8
E07111:	Invalid IRD-View.						
	5.1						
E07113:	Two entity-lists required.						
	5.2	5.3	5.4	5.5			
E07114:	Specified entity-list name already exists.						
	5.6						
W07003:	Specified entity-list is empty.						
	5.1	6.1	8.1.4	8.1.8			
W07005:	List specifies non-existent entities.						
	8.1.3						
W07016:	Entity-list exists and shall be overwritten.						
	5.1	5.2					
W07017:	Duplicate entity-list specified.						
	5.2	5.3	5.4	5.5			
W07018:	Resultant entity-list is empty.						
	5.2	5.3	5.4	5.5			



# Appendix

(This Appendix is not part of American National Standard X3.138-1988, but is included for information only.)

## Collected Syntax

```
build-entity-list-command ::=
    build-word
    entity-list-word
    entity-selection-criteria-clause
    [ list-name-clause ]
    [ using-IRD-views-clause ]
    [ implementor-defined-clauses ]
    ;

delete-selection-option ::=
    entity-name-list
    | using-list-clause
    | entity-selection-criteria-clause

entity-list-1-name ::=
    entity-list-name
    | null-mark

entity-list-2-name ::=
    entity-list-name
    | null-mark

entity-list-difference-command ::=
    difference-word
    ( entity-list-1-name , entity-list-2-name )
    [ = new-entity-list-name ]
    ;

entity-list-intersection-command ::=
    intersect-word
    ( existing-entity-list-names )
    [ = new-entity-list-name ]
    ;

entity-list-subtraction-command ::=
    subtract-word
    ( entity-list-1-name , entity-list-2-name )
    [ = new-entity-list-name ]
    ;

entity-list-union-command ::=
    union-word
    ( existing-entity-list-names )
    [ = new-entity-list-name ]
    ;

entity-name-list ::=
    entity-access-name
    [ , entity-name-list ]

entity-selection-set-spec ::=
    [ using-IRD-views-clause ]
    entity-selection-criteria-clause
    | entity-selection-criteria-clause
    [ using-IRD-views-clause ]
    | using-list-clause

existing-entity-list-name ::=
    entity-list-name
    | null-mark

existing-entity-list-names ::=
    existing-entity-list-name
    [ , existing-entity-list-names ]

IRD-command ::=
    IRD-maintenance-command
    | IRD-output-command
    | IRD-entity-list-command

IRD-entity-list-command ::=
    build-entity-list-command
    | entity-list-union-command
    | entity-list-intersection-command
    | entity-list-difference-command
    | entity-list-subtraction-command
    | name-current-entity-list-command
    | output-entity-list-command
    | output-entity-list-names-command

irds-word ::=
    build-word ::= BUILD
    current-word ::= CURRENT
    difference-word ::= DIFFERENCE
    entity-list-word ::= ENTITY-LIST
    intersect-word ::= INTERSECT
    name-word ::= NAME
    names-word ::= NAMES
    subtract-word ::= SUBTRACT
    union-word ::= UNION
```

## APPENDIX

```
list-name-clause ::=
    list-name-word
    =
    entity-list-name

name-current-entity-list-command ::=
    name-word
    current-word
    entity-list-word
    entity-list-name
    ;

new-entity-list-name ::=
    entity-list-name

output-entity-list-command ::=
    output-word
    entity-list-word
    [ list-name-clause ]
    [ show-title-clause ]
    [ route-clause ]
    ;

output-entity-list-names-command ::=
    output-word
    entity-list-word
    names-word
    [ show-title-clause ]
    [ route-clause ]
    ;

selection-option ::=
    entity-selection-criteria-clause
    | using-list-clause

specified-entities ::=
    using-list-clause
    | name-scan-pattern-list

using-list-clause ::=
    [ using-word ]
    entity-list-word
    [ = entity-list-name ]
```

)

)

)













- X3.113-1987** Programming Language FULL BASIC
- X3.114-1984** Alphanumeric Machines; Coded Character Sets for Keyboard Arrangements in ANSI X4.23-1982 and X4.22-1983
- X3.115-1984** Unformatted 80 Megabyte Trident Pack for Use at 370 tpi and 6000 bpi (General, Physical, and Magnetic Characteristics)
- X3.116-1986** Recorded Magnetic Tape Cartridge, 4-Track, Serial 0.250 Inch (6.30 mm) 6400 bpi (252 bpm), Inverted Modified Frequency Modulation Encoded
- X3.117-1984** Printable/Image Areas for Text and Facsimile Communication Equipment
- X3.118-1984** Financial Services — Personal Identification Number — PIN Pad
- X3.119-1984** Contact Start/Stop Storage Disk, 158361 Flux Transitions per Track, 8.268 Inch (210 mm) Outer Diameter and 3.937 inch (100 mm) Inner Diameter
- X3.120-1984** Contact Start/Stop Storage Disk
- X3.121-1984** Two-Sided, Unformatted, 8-Inch (200-mm), 48-tpi, Double-Density, Flexible Disk Cartridge for 13 262 ftpr Two-Headed Application
- X3.122-1986** Computer Graphics Metafile for the Storage and Transfer of Picture Description Information
- X3.124-1985** Graphical Kernel System (GKS) Functional Description
- X3.124.1-1985** Graphical Kernel System (GKS) FORTRAN Binding
- X3.124.2-1988** Graphical Kernel System (GKS) PASCAL Binding
- X3.125-1985** Two-Sided, Double-Density, Unformatted 5.25-Inch (130-mm), 48-tpi (1.9-tpmm), Flexible Disk Cartridge for 7958 bpr Use
- X3.126-1986** One- or Two-Sided Double-Density Unformatted 5.25-Inch (130-mm), 96 Tracks per Inch, Flexible Disk Cartridge
- X3.127-1987** Unrecorded Magnetic Tape Cartridge for Information Interchange
- X3.128-1986** Contact Start-Stop Storage Disk — 83 000 Flux Transitions per Track, 130-mm (5.118-Inch) Outer Diameter and 40-mm (1.575-Inch) Inner Diameter
- X3.129-1986** Intelligent Peripheral Interface, Physical Level
- X3.130-1986** Intelligent Peripheral Interface, Logical Device Specific Command Sets for Magnetic Disk Drive
- X3.131-1986** Small Computer Systems Interface
- X3.132-1987** Intelligent Peripheral Interface — Logical Device Generic Command Set for Optical and Magnetic Disks
- X3.133-1986** Database Language — NDL
- X3.135-1986** Database Language — SQL
- X3.136-1988** Serial Recorded Magnetic Tape Cartridge for Information Interchange, Four and Nine Track
- X3.137-1988** Unformatted Flexible Disk Cartridge, 90 mm (3.5 Inch) 5.3 tpmm (135 tpi) for 7958 bpr Use
- X3.138-1988** Information Resource Dictionary System (IRDS)
- X3.139-1987** Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC)
- X3.140-1986** Open Systems Interconnection — Connection Oriented Transport Layer Protocol Specification
- X3.141-1987** Data Communication Systems and Services — Measurement Methods for User-Oriented Performance Evaluation
- X3.146-1987** Device Level Interface for Streaming Cartridge and Cassette Tape Drives
- X3.147-1988** Intelligent Peripheral Interface — Device Generic Command Set for Magnetic Tape Drives
- X3.148-1988** Fiber Distributed Data Interface (FDDI) — Token Ring Physical Layer Protocol (PHY)
- X3.153-1987** Open Systems Interconnection — Basic Connection Oriented Session Protocol Specification
- X3.156-1987** Nominal 8-Inch Rigid Disk Removable Cartridge
- X3.157-1987** Recorded Magnetic Tape for Information Interchange, 3200 CPI
- X3.158-1987** Serial Recorded Magnetic Tape Cassette for Information Interchange, 0.150 Inch (3.81 mm), 8000 bpi (315 bpm), Group Code Recording
- X3.162-1988** Two-Sided, High-Density, Unformatted, 5.25-Inch (130-mm), 96 tpi, Flexible Disk Cartridge for 13 262 ftpr Use
- X3.163-1988** Contact Start-Stop Metallic Film Storage Disk—83 333 Flux Transitions per Track, 130-mm (5.118-in) Outer Diameter and 40-mm (1.575-in) Inner Diameter
- X3.165-1988** Programming Language DIBOL
- X11.1-1977** Programming Language MUMPS
- IEEE 416-1978** Abbreviated Test Language for All Systems (ATLAS)
- IEEE 716-1982** Standard C/ATLAS Language
- IEEE 717-1982** Standard C/ATLAS Syntax
- IEEE 770X3.97-1983** Programming Language PASCAL
- IEEE 771-1980** Guide to the Use of ATLAS
- ISO 8211-1986** Specifications for a Data Descriptive File for Information Interchange
- MIL-STD-1815A-1983** Reference Manual for the Ada Programming Language
- NBS-ICST 1-1986** Fingerprint Identification — Data Format for Information Interchange

---

**X3/TRI-82** Dictionary for Information Processing Systems (Technical Report)

# American National Standards for Information Processing

- X3.1-1987** Synchronous Signaling Rates for Data Transmission  
**X3.4-1986** Coded Character Sets — 7-Bit ASCII  
**X3.5-1970** Flowchart Symbols and Their Usage  
**X3.6-1965** Perforated Tape Code  
**X3.9-1978** Programming Language FORTRAN  
**X3.11-1969** General Purpose Paper Cards  
**X3.14-1983** Recorded Magnetic Tape (200 CPI, NRZI)  
**X3.15-1976** Bit Sequencing of the American National Standard Code for Information Interchange in Serial-by-Bit Data Transmission  
**X3.16-1976** Character Structure and Character Parity Sense for Serial-by-Bit Data Communication in the American National Standard Code for Information Interchange  
**X3.17-1981** Character Set for Optical Character Recognition (OCR-A)  
**X3.18-1974** One-Inch Perforated Paper Tape  
**X3.19-1974** Eleven-Sixteenths-Inch Perforated Paper Tape  
**X3.20-1967** Take-Up Reels for One-Inch Perforated Tape  
**X3.21-1967** Rectangular Holes in Twelve-Row Punched Cards  
**X3.22-1983** Recorded Magnetic Tape (800 CPI, NRZI)  
**X3.23-1985** Programming Language COBOL  
**X3.25-1976** Character Structure and Character Parity Sense for Parallel-by-Bit Data Communication in the American National Standard Code for Information Interchange  
**X3.26-1980** Hollerith Punched Card Code  
**X3.27-1987** Magnetic Tape Labels and File Structure  
**X3.28-1976** Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links  
**X3.29-1971** Specifications for Properties of Unpunched Oiled Paper Perforator Tape  
**X3.30-1986** Representation for Calendar Date and Ordinal Date  
**X3.31-1988** Identification of the Counties of the United States  
**X3.32-1973** Graphic Representation of the Control Characters of American National Standard Code for Information Interchange  
**X3.34-1972** Interchange Rolls of Perforated Tape  
**X3.37-1987** Programming Language APT  
**X3.38-1988** Identification of States of the United States (Including the District of Columbia)  
**X3.39-1986** Recorded Magnetic Tape (1600 CPI, PE)  
**X3.40-1983** Unrecorded Magnetic Tape (9-Track 800 CPI, NRZI; 1600 CPI, PE; and 6250 CPI, GCR)  
**X3.41-1974** Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange  
**X3.42-1975** Representation of Numeric Values in Character Strings  
**X3.43-1986** Representations of Local Time of Day  
**X3.44-1974** Determination of the Performance of Data Communication Systems  
**X3.45-1982** Character Set for Handprinting  
**X3.46-1974** Unrecorded Magnetic Six-Disk Pack (General, Physical, and Magnetic Characteristics)  
**X3.47-1988** Identification of Named Populated Places, Primary County Divisions, and Other Locational Entities of the United States  
**X3.48-1986** Magnetic Tape Cassettes (3.81-mm [0.150-Inch] Tape at 32 bpmm [800 bpi], PE)  
**X3.49-1975** Character Set for Optical Character Recognition (OCR-B)  
**X3.50-1986** Representations for U.S. Customary, SI, and Other Units to Be Used in Systems with Limited Character Sets  
**X3.51-1986** Representations of Universal Time, Local Time Differentials, and United States Time Zone References  
**X3.52-1976** Unrecorded Single-Disk Cartridge (Front Loading, 2200 BPI) (General, Physical, and Magnetic Requirements)  
**X3.53-1976** Programming Language PL/I  
**X3.54-1986** Recorded Magnetic Tape (6250 CPI, Group Coded Recording)  
**X3.55-1982** Unrecorded Magnetic Tape Cartridge, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded  
**X3.56-1986** Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded  
**X3.57-1977** Structure for Formatting Message Headings Using the American National Standard Code for Information Interchange for Data Communication Systems Control  
**X3.58-1977** Unrecorded Eleven-Disk Pack (General, Physical, and Magnetic Requirements)  
**X3.60-1978** Programming Language Minimal BASIC  
**X3.61-1986** Representation of Geographic Point Locations  
**X3.62-1987** Paper Used in Optical Character Recognition (OCR) Systems  
**X3.63-1981** Unrecorded Twelve-Disk Pack (100 Megabytes) (General, Physical, and Magnetic Requirements)  
**X3.64-1979** Additional Controls for Use with American National Standard Code for Information Interchange  
**X3.66-1979** Advanced Data Communication Control Procedures (ADCCP)  
**X3.72-1981** Parallel Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded  
**X3.73-1980** Single-Sided Unformatted Flexible Disk Cartridge (for 6631-BPR Use)  
**X3.74-1981** Programming Language PL/I, General-Purpose Subset  
**X3.76-1981** Unformatted Single-Disk Cartridge (Top Loading, 200 tpi 4400 bpi) (General, Physical, and Magnetic Requirements)  
**X3.77-1980** Representation of Pocket Select Characters  
**X3.78-1981** Representation of Vertical Carriage Positioning Characters in Information Interchange  
**X3.79-1981** Determination of Performance of Data Communications Systems That Use Bit-Oriented Communication Procedures  
**X3.80-1988** Interface between Flexible Disk Cartridge Drives and Their Host Controllers  
**X3.82-1980** One-Sided Single-Density Unformatted 5.25-Inch Flexible Disk Cartridge (for 3979-BPR Use)  
**X3.83-1989** ISO Registration According to ISO 2375 — ANSI Sponsorship Procedures  
**X3.84-1981** Unformatted Twelve-Disk Pack (200 Megabytes) (General, Physical, and Magnetic Requirements)  
**X3.85-1981** 1/2-Inch Magnetic Tape Interchange Using a Self Loading Cartridge  
**X3.86-1980** Optical Character Recognition (OCR) Inks  
**X3.88-1981** Computer Program Abstracts  
**X3.89-1981** Unrecorded Single-Disk, Double-Density Cartridge (Front Loading, 2200 bpi, 200 tpi) (General, Physical, and Magnetic Requirements)  
**X3.91M-1987** Storage Module Interfaces  
**X3.92-1981** Data Encryption Algorithm  
**X3.93M-1981** OCR Character Positioning  
**X3.94-1985** Programming Language PANGM  
**X3.95-1982** Microprocessors — Hexadecimal Input/Output, Using 5-Bit and 7-Bit Teleprinters  
**X3.96-1983** Continuous Business Forms (Single-Part)  
**X3.98-1983** Text Information Interchange in Page Image Format (PIF)  
**X3.99-1983** Print Quality Guideline for Optical Character Recognition (OCR)  
**X3.100-1983** Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Packet Mode Operation with Packet Switched Data Communications Network  
**X3.101-1984** Interfaces Between Rigid Disk Drive(s) and Host(s)  
**X3.102-1983** Data Communication Systems and Services — User-Oriented Performance Parameters  
**X3.103-1983** Unrecorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 Inch)  
**X3.104-1983** Recorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in), Phase Encoded  
**X3.105-1983** Data Link Encryption  
**X3.106-1983** Modes of Operation for the Data Encryption Algorithm  
**X3.108-1988** Physical Layer Interface for Local Distributed Data Interfaces to a Nonbranching Coaxial Cable Bus  
**X3.110-1983** Videotex/Teletext Presentation Level Protocol Syntax  
**X3.111-1986** Optical Character Recognition (OCR) Matrix Character Sets for OCR-M  
**X3.112-1984** 14-in (356-mm) Diameter Low-Surface-Friction Magnetic Storage Disk

(Continued on reverse)

January 1989