



TECHNICAL REPORT	EIA-IS-109™	
	Issued 1994-01 Stabilized 2015-07	
CDIF - Transfer Format Syntax - SYNTAX.1		

NOTICE

This document has been taken directly from the original TechAmerica document and contains only minor editorial and format changes required to bring it into conformance with the publishing requirements of SAE Technical Standards. The release of this document is intended to replace the original with the SAE International document. Any numbers established by the original document remain unchanged.

The original document was adopted as an SAE publication under the provisions of the SAE Technical Standards Board (TSB) Rules and Regulations (TSB 001) pertaining to accelerated adoption of specifications and standards. TSB rules provide for (a) the publication of portions of unrevised specifications and standards without consensus voting at the SAE committee level, and (b) the use of the existing specification or standard format.

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2016 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
<http://www.sae.org>

SAE WEB ADDRESS:

Full document details: [SAE EIAIS109](#)

SAE values your input. To provide feedback on this Technical Report, please visit <http://www.sae.org/technical/standards/EIAIS109>

Documents
Document Sharing Hub

Interim Standard

EIA/IS-109

CDIF

CASE Data Interchange Format
A Division of Electronic Industries Association

CDIF
Transfer Format Syntax
SYNTAX.1

January 1994

Electronic Industries Association
CDIF Technical Committee



Documents

Document Sharing Hub

NOTICE

EIA Engineering standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such standards and publications shall not in any respect preclude any member or nonmember of EIA from manufacturing or selling products not conforming to such standards and publications, nor shall the existence of such standards and publications preclude their voluntary use by those other than EIA members, whether the standard is to be used either domestically or internationally.

Recommended standards and publications are adopted by EIA without regard to whether or not their adoption may involve patents on articles, materials or processes. By such action, EIA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standard or publication.

EIA INTERIM STANDARDS

EIA Interim Standards contain information deemed to be of technical value to the industry, and are published at the request of the originating Committee without necessarily following the rigorous public review and resolution of comments which is a procedural part of the development of an EIA Recommended Standard.

EIA Interim Standards should be reviewed on an annual basis by the formulating Committee and a decision made on whether to proceed to develop an EIA Recommended Standard on this subject. EIA Interim Standards must be canceled by the Committee and removed from the EIA Standards Catalog before the end of their third year of existence.

This EIA standard is considered to have international standardization implication, but the International Standardization Organization activity has not progressed to the point where a valid comparison between the EIA standard and the ISO work can be made.

COPYRIGHT 1994

Published by
ELECTRONIC INDUSTRIES ASSOCIATION
Engineering Department
2001 Pennsylvania Avenue N.W.
Washington, D.C. 20006

PRICE: Please refer to the current
Catalog of EIA & JEDEC STANDARDS & ENGINEERING PUBLICATIONS
or contact EIA Electronic Information Group
Phone: (202) 457-4960, FAX: (202) 457-4985

All rights reserved
Printed in U.S.A.

ISBN 0-7908-0016-0

PREFACE

This standard will assist the vendors and users of CASE tools in developing mechanisms for interchanging information between CASE tools. This standard specifies an element of a family of related standards. When used together, these standards specify a mechanism for transferring information between CASE tools.

EIA/IS-106 CDIF - CASE Data Interchange Format - Overview and EIA/IS-107 CDIF - Framework for Modeling and Extensibility should be read first when initially exploring CDIF. The first explains the overall CDIF Architecture and how the family of standards fits together. The second explains the scope, and modeling approach in CDIF. The CDIF Meta-meta-model and extensibility mechanism are also defined in this document.

EIA/IS-110 CDIF - Transfer Format - Encoding ENCODING.1 defines an encoding of SYNTAX.1. ***EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings*** describes how CDIF supports multiple syntaxes and encodings. This standard defines the CDIF Transfer Format syntax, SYNTAX.1.

This standard has been developed with the wide support and participation of vendors, users, academia and government involved in or familiar with the CASE industry, its products and the general requirements associated with interchanging information between these products.

The material contained in this publication was formulated under the cognizance of the CDIF Technical Committee.

CONTENTS

1. SCOPE	1
2. INTRODUCTION.....	3
2.1 Audience	3
2.2 Structure of this Document.....	3
2.3 Conventions	4
2.3.1 Global Conventions.....	4
2.3.2 BNF Conventions.....	5
3. CONCEPTS AND DEFINITIONS.....	7
3.1 Syntax Identifier.....	7
3.2 Token Separation Rules	7
4. SYNTAX SECTIONS AND STRUCTURES IN THE CDIF TRANSFER.....	8
4.1 Introduction.....	8
4.2 CDIF Transfer Components	8
4.2.1 Introduction	8
4.2.2 The Transfer Envelope.....	8
4.2.3 The Transfer Contents.....	8
4.3 Header Section.....	9
4.3.1 Introduction	9
4.3.2 Summary Clause	10
4.4 Meta-model Section.....	11
4.4.1 Introduction	11
4.4.2 CDIF Subject Area Reference Clause.....	12
4.4.3 Meta-model Extension Clause.....	13
4.4.4 Meta-meta-entity Instance	13
4.4.5 Meta-meta-attribute Instance.....	14
4.4.6 Meta-meta-relationship Instance.....	15
4.4.7 Enumerated Meta-attribute Extension	16
4.5 Model Section	16
4.5.1 Introduction	16
4.5.2 Meta-entity Instance.....	17
4.5.3 Meta-relationship Instance	18
4.5.4 Meta-attribute Instance.....	19
4.5.5 Meta-attribute Value	19
4.5.5.1 Introduction	19
4.5.5.2 Bitmap Value.....	19
4.5.5.3 Boolean Value	21
4.5.5.4 Date Value	21
4.5.5.5 Enumerated Value.....	21
4.5.5.6 Float Value	21

4.5.5.7	Identifier Value.....	22
4.5.5.8	Integer Value	22
4.5.5.9	Integer List Value	22
4.5.5.10	Point Value	23
4.5.5.11	Point List Value	24
4.5.5.12	String Value.....	24
4.5.5.13	Text Value	24
4.5.5.14	Time Value	25
4.6	Comments	25
4.7	Syntax Terminal Symbols	26
5.	SYNTAX FORMAL GRAMMAR	28
	APPENDIX A LL(1) EQUIVALENT OF SYNTAX.1	35
	APPENDIX B QUESTIONS AND ANSWERS	43
	GLOSSARY	45
	INDEX.....	53
	RELATED EIA STANDARDS	INSIDE BACK COVER

FIGURES

Figure 1 Position in the CDIF Family of Standards.....	1
Figure 2 CDIF Transfer	9

1. SCOPE

The CDIF Family of Standards is primarily designed to be used as a description of a mechanism for transferring information between CASE tools. It facilitates a successful transfer when the authors of the importing and exporting tools have nothing in common except an agreement to conform to CDIF. The language that is defined for the Transfer Format also has applicability as a general language for Import/Export from repositories. The CDIF Integrated Meta-model defined for CASE also has applicability as the basis of standard definitions for use in repositories.

The standards which form the complete family of CDIF Standards are documented in *EIA/IS-106 CDIF - CASE Data Interchange Format - Overview*. These standards cover the overall framework, the transfer format and the CDIF Integrated Meta-model.

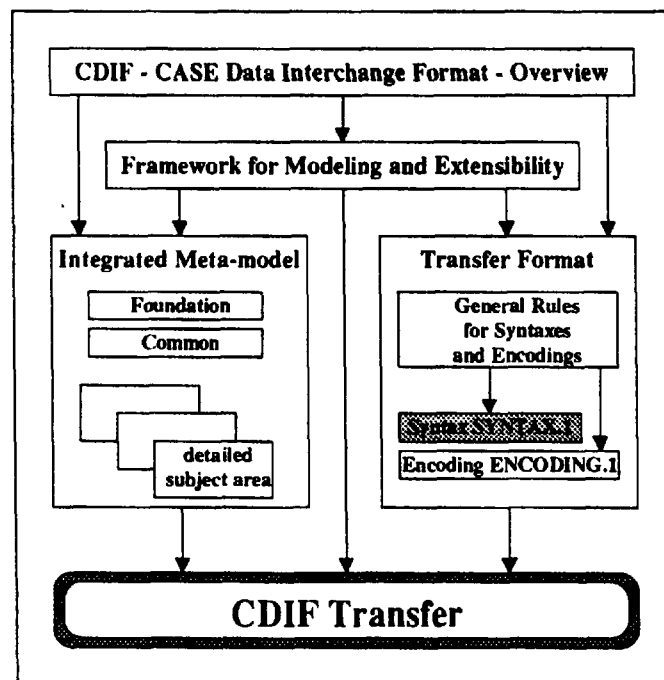


Figure 1
Position in the CDIF Family of Standards

The diagram in Figure 1 depicts the various standards that comprise the CDIF Family of Standards. The shaded box depicts this Standard and the role that it plays in the CDIF Family of Standards.

This document describes the standard CDIF Transfer Syntax. No encodings for SYNTAX.1 are specified in this document. *EIA/IS-110 CDIF - Transfer Format - Encoding ENCODING.1* specifies one standard encoding for this syntax.

2. INTRODUCTION

2.1 Audience

This document is intended to be used by anyone wishing to understand and/or use CDIF. This document provides a definition of a syntax for CDIF transfers. It is suitable for:

- those evaluating CDIF
- those who wish to understand the principles and concepts of a CDIF transfer
- those developing importers and exporters.

The document *EIA/IS-106 CDIF - CASE Data Interchange Format - Overview* and the document *EIA/IS-107 CDIF - Framework for Modeling and Extensibility* should be read first when initially exploring CDIF and before attempting to read other documents in the CDIF Family of Standards.

This document should be read in conjunction with *EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings*.

While there are no specific prerequisites for reading this document, it will be helpful for the reader to have familiarity with the following:

- Entity-Relationship-Attribute modeling
- CASE tools
- Information repositories
- Data dictionaries
- Multiple meta-layer modeling
- Formal grammars
- Transfer formats.

2.2 Structure of this Document

This document is organized into the following sections:

Section 1 'Scope'

This describes the CDIF Family of Standards and positions the current document within the set of documents.

Section 2 'Introduction'

This describes the intended audience, structure and conventions used in the document.

Section 3 'Concepts and Definitions'

This specifies the unique identifier for the syntax defined in this document and describes the concept of token separation used in this document.

Section 4 'Syntax Sections and Structures in the CDIF Transfer'

This section describes, in detail, the exact syntax of each of the three major sections of a transfer.

Section 5 'Syntax Formal Grammar'

This defines the grammar rules for the syntax in Backus Naur Form (BNF).

Appendix A 'LL(1) Equivalent of SYNTAX.1 BNF'

This contains a computer analyzed LL(1) grammar that is equivalent to the BNF in this document.

Appendix B 'Questions and Answers'

This includes detailed responses to questions that have been raised by those reviewing this document. It is intended that the inclusion of these will aid the reader in understanding this document, and related issues concerning CDIF.

Glossary

The Glossary contains definitions of terms used within this Standard.

Index

A comprehensive index is provided.

2.3 Conventions

2.3.1 Global Conventions

The typographical and naming conventions defined below are used throughout the CDIF Family of Standards.

- Double quotes are used to introduce new terms (e.g., "model layer")
- Bold type is used for emphasis (e.g., this is **important**)
- All meta-objects and meta-meta-objects in CDIF (in meta-models and meta-meta-models) are named by concatenating all the words that name the meta-object or meta-meta-object; the first letter of each word is upper-case, the rest are lower-case (e.g., *MetaAttribute*, *AttributeDerivation*, *IsDrawnUsing*, *IsOptional*)

2.3.2 BNF Conventions

The grammar in this document is defined using an extended version of "Backus Naur Form" (BNF).

In BNF, each syntactic element of the language is defined by means of a "production rule". A "production rule" defines a syntactic element in terms of an expression, ordered left to right, that can be used to form an instance of the element. The expression can consist of characters, character strings, and other syntactic elements.

The version of BNF used in this document has the following conventions:

<i>Symbol</i>	<i>Meaning</i>
< >	Angle braces delimit character strings that are the names of syntactic elements, i.e., non-terminal symbols.
::=	This is the definition operator. It is used in a production rule to separate the element defined by the rule from its definition. The element being defined is to the left of the operator and the expression that defines the element is to the right of the operator.
[]	Square braces enclose optional elements in an expression. The portion of the expression within the braces may be explicitly specified or may be omitted.
{ }	Curly braces group elements in an expression, so that the group can be treated as a single syntactic element.
	This is the alternative operator. The vertical bar indicates that the portion of the expression to the right of the bar is an alternative to the portion to the left of the bar. If the vertical bar appears at a position where it is not enclosed in curly or square braces, it specifies a complete alternative for the element defined by the production rule. If the vertical bar appears in a portion of the expression enclosed in curly or square braces, it specifies alternatives for the contents of the innermost pair of such braces.
...	The ellipsis indicates that the element to which it applies in a formula may be repeated any number of times (but note that optionality is covered by the square braces convention). If the ellipsis appears immediately to the right of a closing curly or square brace, "}" or "]", then it applies to the portion of the expression enclosed between that closing brace and the corresponding opening brace, "{" or "[". If an ellipsis appears immediately to the right of any other element, then it applies only to that element.

!!	This is the comment operator. This operator introduces normal English text. It is used as a comment in the BNF and when the definition of a syntactic element is not expressed further in BNF.
<TokenName>	Anything emboldened and surrounded by angle braces is, a token , a BNF non-terminal symbol that is not further defined in the syntax. The detailed representation of a token is specified in the encoding.
Terminal	Anything emboldened, and not surrounded by angle braces, is a literal value (i.e., it represents the character string itself).
x .. y	This is the range operator. It refers to any symbol in the range x to y. For example, 0..255 refers to all integers 0 to 255 inclusive.

The examples in this document use the representation of the terminal symbols defined in the clear text encoding ENCODING.1.

3. CONCEPTS AND DEFINITIONS

3.1 Syntax Identifier

The syntax defined in this version of this standard shall have a CDIF Standardized Syntax Identifier of "SYNTAX.1", and a Version of "02.00.00". This is used to define the standardized syntax used in a CDIF Transfer (see *EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings*).

The syntax of the CDIF Syntax ID and the Syntax Version is as follows:

```
<SyntaxID> ::=  
    "SYNTAX.1"  
  
<SyntaxVersion> ::=  
    "02.00.00"
```

3.2 Token Separation Rules

The "tokens" (or terminal symbols) are the primitive level elements in the syntax. Tokens are normally separated by one or more "whitespace" characters, as defined in the specific encoding. After token identification, these whitespace characters are ignored by importers.

Certain terminal symbols act as token separators; these terminal symbols do not need to be preceded or followed by whitespace characters. All other terminal symbols must be followed by one of these token separators or by one or more whitespace characters. The terminal symbols that act as token separators are listed in Section 4.7 Syntax Terminal Symbols.

4. SYNTAX SECTIONS AND STRUCTURES IN THE CDIF TRANSFER

4.1 Introduction

This section describes, in detail, the exact syntax of each of the major sections of a transfer, together with the syntax of each of the component structures.

Examples of transfers given in this section use the CDIF Standard Encoding as defined in *EIA/IS-110 CDIF - Transfer Format - Encoding ENCODING.1*. Two complete examples of a transfer are contained in that document.

4.2 CDIF Transfer Components

4.2.1 Introduction

The general syntax of a CDIF Transfer is as follows:

```
<CDIFTransfer> ::=  
    <TransferEnvelope>  
    <TransferContents>
```

Figure 2 illustrates the complete structure of a CDIF transfer. A transfer is typically contained in a file, but other means of transfer such as pipes, mailboxes, shared memory, or any other mechanism that can be interpreted as a byte stream may be used.

4.2.2 The Transfer Envelope

The Transfer Envelope consists of the CDIF Signature, the Syntax Identifier and the Encoding Identifier. As the Transfer Envelope syntax is the same for any CDIF transfer, it is defined in *EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings*. The syntax identifier for SYNTAX.1 is specified in 3.1 Syntax Identifier of this document.

4.2.3 The Transfer Contents

As the first level of the grammar of the Transfer Contents is the same for any CDIF transfer, it is defined in *EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings*, but is reproduced here for clarity:

```
<TransferContents> ::=  
    <HeaderSectionClause> <MetaModelSectionClause>  
    [ <ModelSectionClause> ]
```

The further levels are detailed in the next section.

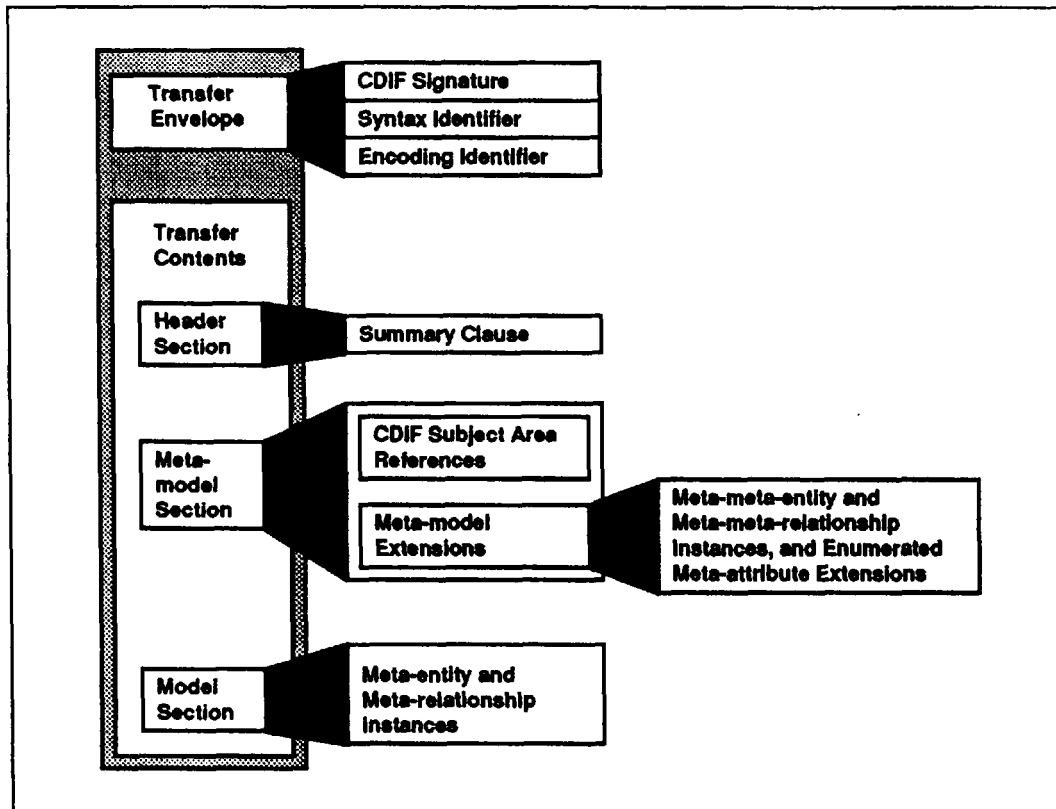


Figure 2
CDIF Transfer

4.3 Header Section

4.3.1 Introduction

The Header Section defines information that applies to the whole transfer.

The syntax of the Header Section clause is:

```
<HeaderSectionClause> ::=
    <OpenScope> <HeaderKeyword>
    <SummaryClause>
    <CloseScope>
```

that is, an introductory keyword followed by a Summary clause, all enclosed within scope delimiters.

An example of the Header Section clause is:

```
(:HEADER
  <SummaryClause>
)
```

4.3.2 Summary Clause

The Summary clause specifies summary information about the transfer, in the form of a number of items. Each item is introduced by an identifier. This standard defines meanings for certain summary section identifiers. Other identifiers may be used, to introduce other items of summary information, but their meanings are not defined in this standard.

The syntax of the Summary clause is:

```
<SummaryClause> ::=
    <OpenScope> <SummaryKeyword>
    [ <IdentifierValuePair> ] ...
    <CloseScope>

<IdentifierValuePair> ::=
    <OpenScope>
    <SummaryIdentifier> <StringValue>
    <CloseScope>

<SummaryIdentifier> ::=
    <Identifier>

<StringValue> ::= <String>
```

The identifiers must conform to the rules for data type Identifier.

The defined summary section identifiers and their definitions are:

- ExporterName - The name of the exporting software product
- ExporterVersion - The version of the exporting software product
- ExportDate - The date of commencement of the export, in the form YYYY/MM/DD
- ExportTime - The time of commencement of the export, in the form HH:MM:SS
- PublisherName - The name of the person (or function) who initiated the export

An example of a Summary clause is:

```
(:SUMMARY
  (ExporterName      "AutoPal")
  (ExporterVersion   "3.2")
  (ExportDate        "1991/01/12")
  (ExportTime        "13:00:00")
  (PublisherName     "Andrew Shilling")
  (HardwarePlatform  "Apple MacIntosh")
  (VendorCompanyName "UltraCASE")
)
```

4.4 Meta-model Section

4.4.1 Introduction

The Meta-model Section of the transfer consists of references to standardized Subject Areas, followed by extensions to the Meta-model. The Meta-model for the transfer, known as the "Working Meta-model", is defined by the set of meta-meta-entity and meta-meta-relationship instances that are used in any of the referenced Subject Areas, plus those added by extensions.

The Working Meta-model may be extended in the following ways:

1. Adding instances to the Working Meta-model of meta-meta-entities to define additional *MetaEntities*, *MetaRelationships*, *MetaAttributes* and *SubjectAreas*.
2. Adding instances to the Working Meta-model of meta-meta-relationships to define additional associations between instances of meta-meta-entities in the Working Meta-model. The following are the meta-meta-relationships currently defined in *EIA/IS-107 CDIF - Framework for Modeling and Extensibility*:
 - *HasSource* to identify the source *MetaEntity* of a *MetaRelationship*
 - *HasDestination* to identify the destination *MetaEntity* of a *MetaRelationship*
 - *IsLocalMetaAttributeOf* to associate a *MetaAttribute* with a specific *MetaEntity* or *MetaRelationship*
 - *HasSubtype* to define that one *MetaEntity* or *MetaRelationship* is a subtype of another
 - *IsUsedIn* to define that a *MetaEntity*, *MetaRelationship* or *MetaAttribute* is used in a specific *SubjectArea*.
3. Extending domains of *MetaAttributes* with Enumerated data type to include additional legal values.

As a conveyor of meta-models, this section contains instances of meta-meta-model concepts. The mapping between these concepts, as defined in *EIA/IS-107 CDIF - Framework for Modeling and Extensibility*, and this syntax is straight forward and therefore clarification is included only where necessary.

The syntax of the Meta-model Section clause is:

```
<MetaModelSectionClause> ::=
    <OpenScope> <MetaModelKeyword>
    <CDIFSubjectAreaReferenceClause>...
    [ <MetaModelExtensionClause> ]...
    <CloseScope>
```

An example of the Meta-model Section clause is:

```
(:META-MODEL
    <CDIFSubjectAreaReferenceClause>...
    [ <MetaModelExtensionClause> ]...
)
```

4.4.2 CDIF Subject Area Reference Clause

The syntax of the CDIF Subject Area Reference clause is:

```
<CDIFSubjectAreaReferenceClause> ::=
    <OpenScope>
    <SubjectAreaReferenceKeyword> <SubjectAreaName>
    <OpenScope>
    <VersionNumberKeyword> <SubjectAreaVersionNumber>
    <CloseScope>
    <CloseScope>

<SubjectAreaName> ::=
    <MetaObjectName>

<SubjectAreaVersionNumber> ::=
    <String>
```

<SubjectAreaName> must be the value of the *Name* meta-meta-attribute of the relevant *SubjectArea* instance. <SubjectAreaVersionNumber> must be the value of the *VersionNumber* meta-meta-attribute of the relevant *SubjectArea* instance. A reference to at least one CDIF Subject Area must be included in the Meta-model Section.

An example of the CDIF Subject Area Reference clause is:

```
(:SUBJECTAREAREFERENCE Foundation
    (:VERSIONNUMBER "01.00")
)
```

4.4.3 Meta-model Extension Clause

The syntax of the Meta-model Extension clause is:

```
<MetaModelExtensionClause> ::=  
    <MetaMetaEntityInstance>  
    | <MetaMetaRelationshipInstance>  
    | <EnumeratedMetaAttributeExtension>
```

Note that although the syntax allows Meta-meta-entity Instance, Meta-meta-relationship Instance and Enumerated Meta-attribute Extension clauses to be placed in arbitrary order, no forward references are allowed in a transfer.

4.4.4 Meta-meta-entity Instance

Instances of meta-meta-entities are specified by the name of the meta-meta-entity (e.g., *MetaEntity*, *MetaRelationship*, *MetaAttribute* or *SubjectArea*), followed by a unique CDIF Meta-identifier for the instance, followed by the names and values of the other meta-meta-attributes for the meta-meta-entity.

The syntax of the Meta-meta-entity Instance clause is:

```
<MetaMetaEntityInstance> ::=  
    <OpenScope>  
    <MetaMetaEntityName>  
    <CDIFMetaIdentifier>  
    [ <MetaMetaAttributeInstance> ] ...  
    <CloseScope>  
  
<MetaMetaEntityName> ::=  
    <MetaMetaObjectName>  
  
<CDIFMetaIdentifier> ::=  
    <Identifier>
```

<MetaMetaEntityName> must be the name of a meta-meta-entity, as defined in *EIA/IS-107 CDIF - Framework for Modeling and Extensibility*, e.g., *MetaEntity*, *MetaRelationship*, *MetaAttribute* or *SubjectArea*.

CDIF Meta-identifiers beginning with a numeric character are reserved for use in the CDIF Integrated Meta-model.

A meta-meta-attribute Instance clause is used to represent each of the meta-meta-attributes (other than the CDIFMetaIdentifier meta-meta-attribute) of the meta-meta-entity.

<MetaMetaAttributeInstance> is defined in the next section.