

BS ISO 15889:2011



BSI Standards Publication

Space data and information transfer systems — The data description language EAST specification

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO 15889:2011. It supersedes BS ISO 15889:2003 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee ACE/68/-/7, Space systems and operations - Space data and information transfer systems.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© BSI 2011

ISBN 978 0 580 74129 6

ICS 49.140

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2011.

Amendments issued since publication

Date	Text affected
------	---------------

INTERNATIONAL STANDARD

BS ISO 15889:2011

ISO
15889

Third edition
2011-10-15

Space data and information transfer systems — The data description language EAST specification

*Systèmes de transfert des informations et données spatiales —
Spécification du langage de description de données EAST*



Reference number
ISO 15889:2011(E)
Documents
Document Sharing Hub
© ISO 2011



COPYRIGHT PROTECTED DOCUMENT

© ISO 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

This third edition cancels and replaces the second edition (ISO 15889:2003), which has been technically revised.

ISO 15889 was prepared by the Consultative Committee for Space Data Systems (CCSDS) as CCSDS 644.0-B-3, June 2010 and was adopted without modifications except those stated in Clause 2 of this International Standard by Technical Committee ISO/TC 20, *Aircraft and space vehicles*, Subcommittee SC 13, *Space data and information transfer systems*.

Space data and information transfer systems — The data description language EAST specification

1 Scope

This International Standard defines the Enhanced Ada SubseT (EAST) language used to create descriptions of data called Data Description Records (DDRs). Such DDRs ensure a complete and exact understanding of the data and allow it to be interpreted in an automated fashion. This means that a software tool is able to analyse a DDR and interpret the format of the associated data. This allows the software to extract values from the data on any host machine (i.e. on a different machine from the one that produced the data).

The scope and field of application are furthermore detailed in subclauses 1.1 and 1.2 of the enclosed CCSDS publication.

2 Requirements

Requirements are the technical recommendations made in the following publication (reproduced on the following pages), which is adopted as an International Standard:

CCSDS 644.0-B-3, June 2010, *The data description language EAST specification*.

For the purposes of international standardization, the modifications outlined below shall apply to the specific clauses and paragraphs of publication CCSDS 644.0-B-3.

Pages i to vi

This part is information which is relevant to the CCSDS publication only.

Page 1-5

Add the following information to the reference indicated:

- [1] ISO 8859-1:1987 has been withdrawn and replaced by ISO/IEC 8859-1:1998.
- [2] ISO/IEC 10646-1:1993 has been withdrawn and has been replaced by ISO/IEC 10646:2011.

Page E-1

Add the following information to the reference indicated:

- [E2] Document CCSDS 620.0-B-2, May 1992, is equivalent to ISO 12175:1994.

3 Revision of publication CCSDS 644.0-B-3

It has been agreed with the Consultative Committee for Space Data Systems that Subcommittee ISO/TC 20/SC 13 will be consulted in the event of any revision or amendment of publication CCSDS 644.0-B-3. To this end, NASA will act as a liaison body between CCSDS and ISO.

(Blank page)



Recommendation for Space Data System Standards

THE DATA DESCRIPTION LANGUAGE EAST SPECIFICATION (CCSD0010)

RECOMMENDED STANDARD

CCSDS 644.0-B-3

BLUE BOOK
June 2010

(Blank page)

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

AUTHORITY

Issue:	Recommended Standard, Issue 3
Date:	June 2010
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS Recommendations is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
 - The **standard** itself.
 - The anticipated date of initial operational capability.
 - The anticipated duration of operational service.
- o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

<http://www.ccsds.org/>

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- Russian Federal Space Agency (RFSA)/Russian Federation.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
- Danish National Space Center (DNSC)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

- United States Geological Survey (USGS)/USA.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

DOCUMENT CONTROL

Document	Title and Issue	Date	Status
CCSDS 644.0-B-1	Recommendation for Space Data System Standards: The Data Description Language EAST Specification (CCSD0010), Issue 1	May 1997	Original issue, superseded.
CCSDS 644.0-B-2	Recommendation for Space Data System Standards: The Data Description Language EAST Specification (CCSD0010), Issue 2	November 2000	Issue 2, superseded: – extends EAST ability to handle repeated data items where repetition is terminated by a marker.
CCSDS 644.0-B-3	The Data Description Language EAST Specification (CCSD0010), Recommended Standard, Issue 3	June 2010	Current issue: – adds requirement to specify EAST version.

NOTE – Substantive changes from the previous issue are indicated by change bars in the inside margin.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

CONTENTS

<u>Section</u>	<u>Page</u>
1 INTRODUCTION	1-1
1.1 PURPOSE AND SCOPE	1-1
1.2 APPLICABILITY	1-1
1.3 RATIONALE	1-1
1.4 DOCUMENT STRUCTURE	1-2
1.5 DEFINITIONS	1-2
1.5.1 TERMS	1-2
1.5.2 NOMENCLATURE	1-2
1.5.3 CONVENTIONS	1-3
1.6 REFERENCES	1-5
2 OVERVIEW	2-1
2.1 DESIGN AIMS	2-1
2.2 STRUCTURE OF AN EAST DESCRIPTION	2-1
2.3 LANGUAGE SUMMARY	2-2
3 DEFINITION OF THE EAST LANGUAGE	3-1
3.1 LEXICAL ELEMENTS	3-1
3.1.1 SEPARATORS AND DELIMITERS	3-1
3.1.2 COMMENTS	3-1
3.1.3 IDENTIFIERS	3-2
3.1.4 NUMERIC LITERALS	3-2
3.2 LOGICAL DESCRIPTION	3-7
3.2.1 TYPE DECLARATIONS	3-8
3.2.2 SUBTYPE DECLARATIONS	3-28
3.2.3 OBJECT DECLARATIONS	3-31
3.2.4 REPRESENTATION CLAUSES	3-35
3.3 PHYSICAL DESCRIPTION	3-47
3.3.1 STORING ARRAYS	3-48
3.3.2 STORING OCTETS/BITS	3-48

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

CONTENTS (continued)

<u>Section</u>	<u>Page</u>
3.3.3 REPRESENTATION OF SCALAR TYPES.....	3-50
3.3.4 RELATIONSHIP BETWEEN THE REPRESENTATION OF SCALAR TYPES AND LOGICAL TYPES	3-62
3.3.5 TEMPLATE OF A PHYSICAL DESCRIPTION PART	3-64
4 RESERVED KEYWORDS	4-1
5 CONFORMANCE	5-1
ANNEX A ACRONYMS AND GLOSSARY	A-1
ANNEX B CHARACTER DEFINITION	B-1
ANNEX C EAST FORMAL SYNTAX SPECIFICATION	C-1
ANNEX D MAIN DIFFERENCES BETWEEN ADA AND EAST	D-1
ANNEX E INFORMATIVE REFERENCES	E-1
INDEX	I-1

Figure

1-1 Example of Syntax Diagram.....	1-3
3-1 Identifier Definition Diagram	3-2
3-2 Decimal Literal Definition Diagram.....	3-2
3-3 Integer Decimal Literal Definition Diagram.....	3-3
3-4 Real Decimal Literal Definition Diagram.....	3-3
3-5 Integer Definition Diagram.....	3-3
3-6 Exponent Definition Diagram.....	3-4
3-7 Based Literal Definition Diagram.....	3-4
3-8 Integer Based Literal Definition Diagram.....	3-5
3-9 Real Based Literal Definition Diagram	3-5
3-10 Based Integer Definition Diagram	3-5
3-11 Integer Literal Definition Diagram	3-6
3-12 Real Literal Definition Diagram	3-6
3-13 Logical Part Structure	3-8
3-14 Enumeration Type Specification Diagram.....	3-9
3-15 Enumeration Literal Definition Diagram.....	3-9
3-16 Integer Type Specification Diagram	3-10
3-17 Real Type Specification Diagram	3-11
3-18 Array Type Specification Diagram	3-12
3-19 Index Specification Diagram	3-13
3-20 Record Type Specification Diagram.....	3-15
3-21 Component Declaration Diagram	3-15
3-22 Default Value Definition Diagram.....	3-16
3-23 Index Constraint Diagram.....	3-17

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

CONTENTS (continued)

<u>Figure</u>	<u>Page</u>
3-24 Discriminant Specification Diagram.....	3-18
3-25 Variant Part Specification Diagram.....	3-19
3-26 Discriminants in a Packet Format.....	3-22
3-27 Actual Discriminant Value Declaration Diagram.....	3-26
3-28 Type Summary.....	3-27
3-29 Subtype Declaration Diagram.....	3-28
3-30 Enumeration Constraint Diagram.....	3-28
3-31 Integer Constraint Diagram.....	3-29
3-32 Real Constraint Diagram.....	3-30
3-33 Variable Declaration Diagram.....	3-31
3-34 Constant Declaration Diagram.....	3-32
3-35 Length Clause Specification Diagram.....	3-36
3-36 Enumeration Clause Specification Diagram.....	3-38
3-37 Component Representation Clause Specification Diagram.....	3-39
3-38 Record Representation Clause Specification Diagram.....	3-39
3-39 First Tree Structure.....	3-41
3-40 Second Tree Structure.....	3-42
3-41 Third Tree Structure.....	3-43
3-42 Fourth Tree Structure.....	3-44
3-43 Distance Specification Diagram.....	3-46
3-44 Record Value Specification Diagram.....	3-54
3-45 Component Value Definition Diagram.....	3-54
3-46 Array Value Specification Diagram.....	3-55
3-47 ASCII Encoded Decimal Integer Format.....	3-60
3-48 ASCII Encoded Decimal Real Format.....	3-61

Example

1-1 Example of BNF.....	1-4
3-1 Decimal Literals.....	3-4
3-2 Based Literals.....	3-6
3-3 Enumeration Type Declarations.....	3-10
3-4 Integer Type Declarations.....	3-10
3-5 Real Type Declarations.....	3-11
3-6 Constrained Array Type Definitions.....	3-13
3-7 Unconstrained Array Type Definitions.....	3-14
3-8 Record Type Definitions.....	3-18
3-9 Record Type Definition with Discriminant.....	3-20
3-10 Record Type Definition with Discriminant.....	3-20
3-11 Logical Description of the Packet Format.....	3-24

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

CONTENTS (continued)

<u>Example</u>	<u>Page</u>
3-12 Calculated Size Array	3-25
3-13 Calculated Component Presence Condition	3-26
3-14 Character Declarations	3-29
3-15 Subtype Declarations	3-30
3-16 Variable Declaration	3-31
3-17 Constant Declaration	3-32
3-18 Number Declarations	3-33
3-19 Marker Declaration	3-34
3-20 EOF Marker Declaration	3-34
3-21 Marker Declaration in Record Definition	3-35
3-22 Length Clause Declarations	3-36
3-23 Explicit Description of Unused Space	3-37
3-24 Enumeration Clause Declarations	3-38
3-25 Type Definitions	3-40
3-26 Complete Record Representation Clause Declaration	3-41
3-27 Incomplete Record Representation Clause Declaration	3-42
3-28 Complete Record Representation Clause Declaration	3-43
3-29 Complete Record Representation Clause Declaration	3-45
3-30 Record Representation Clause Using WORD_32_BITS	3-46
3-31 Actual Array Storage Method	3-48
3-32 Octet Storage Possibilities	3-49
3-33 Actual Bit Order	3-50
3-34 Bit Ordering	3-52
3-35 Bit Ordering for the Above 16-Bit Signed Integer	3-55
3-36 Actual Binary Representation of the Above 16-Bit Signed Integer	3-55
3-37 Bit Ordering for the Above 16-Bit Unsigned Integer	3-56
3-38 Actual Binary Representation of the Above 16-Bit Unsigned Integer	3-56
3-39 Bit Ordering for the Above 32-Bit Real	3-57
3-40 Actual Binary Representation of a 32-Bit Real	3-57
3-41 ASCII Enumeration Type Logical Declaration	3-59
3-42 ASCII Enumeration Type Physical Description	3-60
3-43 ASCII Integer Type Logical Declaration	3-62
3-44 ASCII Integer Type Physical Description	3-62
3-45 ASCII Real Type Logical Declaration	3-62
3-46 ASCII Real Type Physical Description	3-62

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

The purpose of this document is to establish a common Recommendation for the specification of a standard language for describing and expressing data in order to interchange them in a more uniform and automated fashion within and among Agencies participating in the Consultative Committee for Space Data Systems (CCSDS).

This Recommendation defines the Enhanced Ada SubseT (EAST) language used to create descriptions of data, called Data Description Records (DDRs). Such DDRs ensure a complete and exact understanding of the data and allow it to be interpreted in an automated fashion. This means that a software tool is able to analyze a DDR and interpret the format of the associated data. This allows the software to extract values from the data on any host machine (i.e., on a different machine from the one that produced the data).

A first look at reference [E4], which is a tutorial for the EAST language, may aid the user in understanding this document. Reference [E4] describes the requirements, explains how to use the EAST language to describe non-ambiguous data, and suggests practices and tools to the users.

This Recommendation is registered under the CCSDS Authority and Description Identifier (ADID): CCSD0010.

1.2 APPLICABILITY

The specifications in this document may be applicable to all space-related science and engineering data exchanges where data descriptions are desired, and these descriptions need to provide an unambiguous description of the record structure of the data.

1.3 RATIONALE

The Consultative Committee for Space Data Systems has defined the Standard Formatted Data Unit (SFDU) concept for the implementation of standard data structures to be used for the interchange of data within and among space agencies.

SFDU data products may be viewed as containing application data (that is the data which is of primary interest, e.g., actual measurements) and data description information (that is the information telling how the application data are formatted).

The data description information shall be provided in a form that is understandable by the agencies involved in the data interchange. That is the reason why the CCSDS must provide some recommendations for the definition of standard description languages. EAST is one of the recommended languages.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

1.4 DOCUMENT STRUCTURE

The Recommendation is structured as follows:

- Section 2 provides an overview of the EAST language.
- Section 3 specifies the EAST language and defines its usage in Data Descriptions.
- Section 4 lists the EAST reserved keywords.
- Annex A contains acronyms and the glossary of terms used in this document.
- Annex B defines the character set to be used in an EAST data description, as well as a predefined type called CHARACTER.
- Annex C provides the EAST formal specification using a simple variant of the Backus-Naur-Form (BNF).
- Annex D lists the main differences between the Ada programming language and EAST.
- Annex E lists the informative references.

1.5 DEFINITIONS

1.5.1 TERMS

The terms used throughout this document are listed in annex A. They are also explained in the text when they are first used.

1.5.2 NOMENCLATURE

The following conventions apply throughout this Recommendation:

- a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;
- b) the word ‘should’ implies an optional, but desirable, specification;
- c) the word ‘may’ implies an optional specification;
- d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

The syntax of the language is described using a simple variant of Backus-Naur-Form with the following conventions:

- a) Boldface words are used to denote reserved keywords.
- b) Square brackets enclose optional items.
- c) Braces enclose a repeated item. This item may appear zero or more times.
- d) A vertical bar separates alternative items unless it occurs immediately after an opening brace ({}): in this case it represents the character ‘vertical bar’.
- e) If the name of any syntactic category starts with an italicized part, it is equivalent to the category name without the italicized part. The italicized part is intended to convey some semantic information. This facility used for the BNF intends to assimilate every element like *<italicized_part_name>* to the previously defined element *<name>*.

The following example presents the definition of Item A using a simple variant of BNF. Item A is defined as first a keyword (‘type’), then followed by an italicized Item B (already defined, and known as Item B), then followed by a keyword (‘is’) and a delimiter (‘(’). The structure is followed by a choice. The choice may be repeated any number of times, each time a delimiter (‘,’) is inserted. The structure is ended by two delimiters (‘)’ and ‘;’). The choice is between Items B and C.

```
<Item A> ::= type <Italicized_Item B> is ( <choice> { , <choice> } ) ;  
<choice> ::= <Item B> | <Item C>
```

Example 1-1: Example of BNF

In the case of any confusion, the syntax diagram and the associated text are always the reference for the EAST syntax, and not the BNF.

This document uses examples to illustrate the EAST. The following conventions are used in the examples:

- a) bold characters denote reserved keyword or delimiters;
- b) user type names or user variable names are provided using uppercase letters, although EAST is not a case-sensitive language.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

1.6 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommended Standard. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommended Standards.

- [1] *Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 1: Latin Alphabet No. 1*. International Standard, ISO 8859-1:1987. Geneva: ISO, 1987.
- [2] *Information Processing—Universal Multiple-Octet Coded Character Set (UCS)*. International Standard, ISO/IEC 10646-1:1993.

(Blank page)

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

2 OVERVIEW

2.1 DESIGN AIMS

EAST was designed with three overriding concerns: data description capabilities, human readability, and computer interpretability.

The need for data description languages that supply complete and non-ambiguous information about the format and the nature of the described data is well established.

Any user must be able to understand descriptions of data, with a minimal effort. Error-prone notations have been avoided, and the syntax of the EAST language avoids the use of cryptic forms in favor of more English-like constructs.

EAST is a formal language and not a natural language: it is a machine compilable (or interpretable) language. The formal nature of EAST allows the control of data descriptions and the interpretation of data in an automated fashion.

2.2 STRUCTURE OF AN EAST DESCRIPTION

An EAST Data Description Record (DDR) includes a syntactic, and in some way semantic, description of the data called a logical description, which is followed by a physical description. The physical description makes possible the interpretation of the actual bit patterns encountered on the medium. Each description part of a DDR consists of an EAST unit, called a package: one for the logical part and another one for the associated physical part.

The logical part of an EAST description includes:

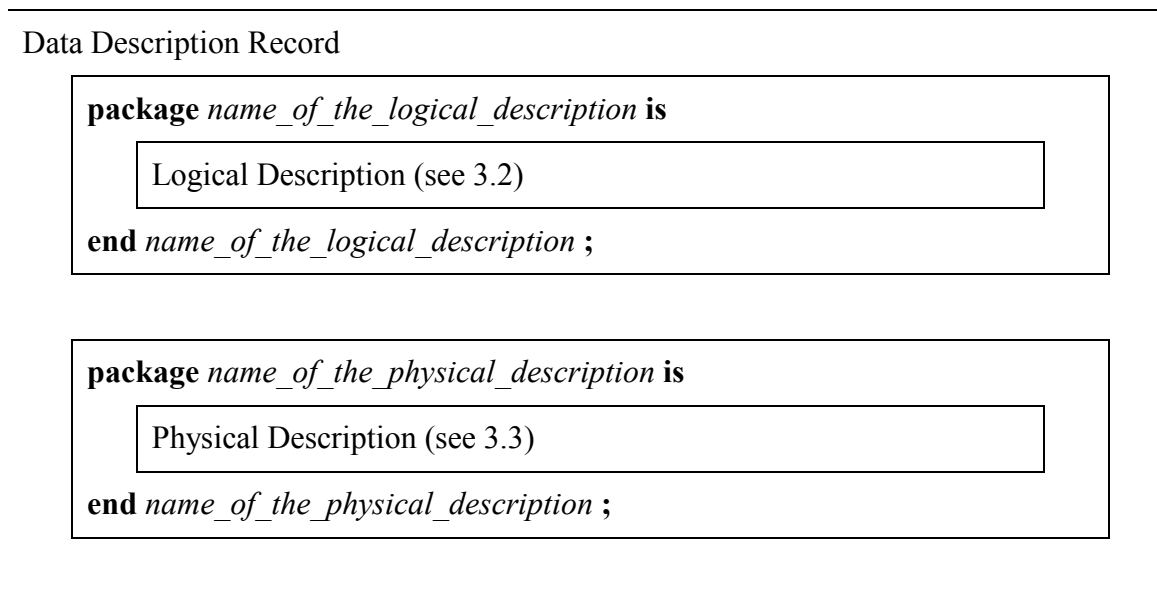
- a logical description of all components of the exchanged data (see 3.2.1 and 3.2.2);
- their size in bits (see 3.2.4.1);
- their location within the set of the described data (see 3.2.4.3).

The physical part of an EAST description includes:

- the representation of some basic data types (enumeration, integer, and real) defined in the logical description and dependent on the machine that has generated the data (see 3.3.3);
- the array organization (first-index-first or last-index-first) used by the generating machine (see 3.3.1);
- the octet and bit organization on the medium (high-order-first or low-order-first—see 3.3.2).

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

A DDR created using the EAST Language has the following structure:



The logical description always precedes the physical description. The logical and the physical packages are mandatory even if the content of the physical one can be empty (see 3.3).

The two part design of the DDR is intended to allow interchangeable physical description parts for one logical description part, provided that the length of fields in bits in the logical description are supported by field lengths of the same number of bits in the physical description part. For example, a 32 bit real number on a IEEE architecture has a physical description different from the one on a 1750 architecture, although lengths in bits of each field are equal. Note that the representations written to an exchange medium do not have to be the ones typically supported by the writing machine.

The data block associated with the DDR contains one or more complete sets of data. The DDR describes a single set only and is repetitively applied to fully interpret the data block.

2.3 LANGUAGE SUMMARY

An EAST description is composed of two units, called packages. The first one is a *logical* description and the second one is a *physical* description of the data. The logical part of an EAST description provides syntactic information and in some way semantic information, i.e., the information needed by a user to understand the data he has to deal with. The physical part of an EAST description provides a bit-level description that ensures the non-ambiguous interpretation of the data.

The syntax used in each of the two packages is based on the type and object concept. A type is a model, defined once, that is used to create many occurrences (objects) of the models.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

Every data item described in an EAST description is an object. An object in the language has a type, which characterizes a set of values. The basic classes of types are *scalar types* (comprising *enumeration* and *numeric types*, describing single elements), and *composite types* (comprising *array* and *record types*, describing sequences of objects).

A type has a name: if well chosen, this name is a way to provide the meaning of the model (e.g., the type DATE may describe a CCSDS date). An object has a name also: this name is a way to provide (if any) the particularity of the occurrence (e.g., the object DATE_AT_THE_BEGINNING_OF_THE_ORBIT of the type DATE may represent a particular date). The name used to identify a type or an object can be any identifier except for an EAST reserved keyword (reserved keywords are provided in section 4).

An enumeration type defines an ordered set of distinct enumeration literals; for example, a Boolean type defines two enumeration literals (TRUE and FALSE). The enumeration type CHARACTER is predefined and given in 3.2.1.1.

Numeric types provide a means of describing whole numbers and real numbers. Whole numbers are described using integer types. Real numbers are described using floating point types, with relative bounds on the error.

Composite types allow definitions of structured objects with related components. The composite types of the EAST language are arrays and records. An array is an object with indexed components of the same type. The array type STRING is predefined and given in 3.2.1.1. A record is an object with named components of possibly different types.

A record may have special components called *discriminants*. Discriminants specify either which of alternative record structures is to be used or the dynamic size of an internal array (depending on the values of the discriminants).

The concept of type is refined by the concept of *subtype*, whereby a user can constrain the set of allowed values of a type. Subtypes can be used to define subranges of scalar types and arrays with a limited set of index values.

Representation clauses are used to specify the mapping between logical types and their physical representations. For example, the user specifies that objects of a given type are represented with a given number of bits, or the components of a record are represented using a given storage layout.

NOTES

- 1 EAST is a subset of the Ada programming language (reference [E3]). EAST contains therefore most of the declarative features of Ada, but no algorithmic features.
- 2 The declarative part of Ada normally defines the logical entities and sometimes some of their physical characteristics. EAST extends the descriptive power of the Ada language (using conventions in the physical packages). It is able to describe not only the logical aspects of a data item, but also all its physical aspects.

(Blank page)

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

3 DEFINITION OF THE EAST LANGUAGE

An EAST Data Description is a text composed of lexical elements, each composed of ASCII characters: the 128 first characters of the Latin Alphabet No. 1 (see reference [1] and/or annex B). The rules of composition are given in 3.1. They are applicable to the whole EAST DDR.

3.1 LEXICAL ELEMENTS

A **lexical element** is either a delimiter, an identifier (which may be a reserved word), a numeric literal, a character string, a string literal, or a comment. The rules of composition are given in this section.

3.1.1 SEPARATORS AND DELIMITERS

In some cases an explicit separator is required to separate adjacent lexical elements (namely, when without separation, interpretation as a single lexical element is possible). A **separator** is any of a space character, a control character, or the end of a line.

- A space character is a separator except within a comment, a string literal, or a space character literal.
- Control characters other than horizontal tabulation are always separators. Horizontal tabulation is a separator except within a comment.
- The end of a line is always a separator. What defines the end of a line is specified in annex B.

A **delimiter** is either one of the following special characters:

& ' () * + , - . / : ; < = > |

or one of the following compound delimiters, each composed of two adjacent special characters:

=> .. ** := /= >= <= << >> <>

Each of the special characters listed for single character delimiters is a single delimiter except if this character is used as a character of a compound delimiter, or as a character of a comment, string literal, character literal, or numeric literal.

The remaining forms of lexical elements are described in 3.1.2, 3.1.3 and 3.1.4.

3.1.2 COMMENTS

A comment starts with two adjacent hyphens and extends up to the end of the line. A comment can appear on any line of a description.

3.1.3 IDENTIFIERS

Identifiers are used as names and also as reserved words. See figure 3-1 for the lexical definition of an identifier:

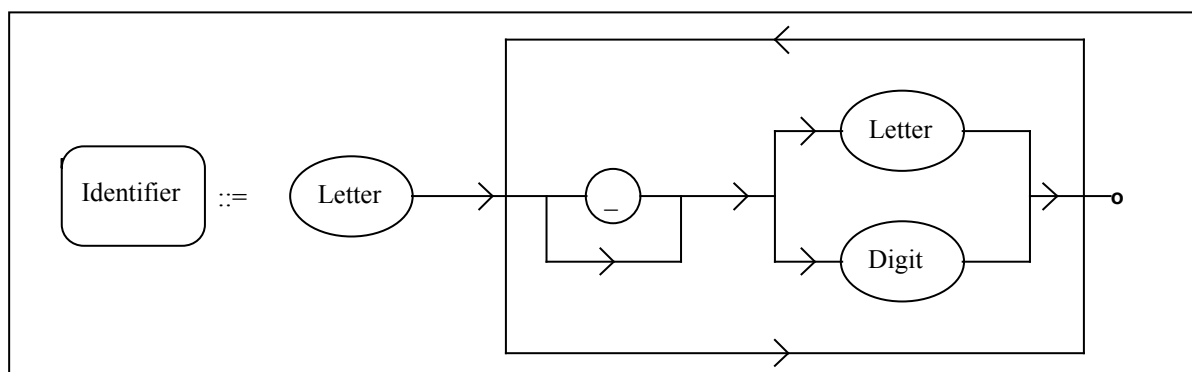


Figure 3-1: Identifier Definition Diagram

All characters of an identifier are significant, including any underline character inserted between a letter or a digit and an adjacent letter or digit. Identifiers differing in the use of corresponding upper and lower case letters are considered to be the same.

3.1.4 NUMERIC LITERALS

A numeric literal is either a decimal literal or a based literal. A decimal literal is a numeric literal expressed in the conventional decimal notation (that is, the base is implicitly ten). A based literal is a numeric literal expressed in a form that specifies the base explicitly. The base can only be either two, eight, or sixteen.

In another way, a numeric literal is either an integer literal (decimal or based) or a real literal (decimal or based). See figure 3-2.

a) decimal literals

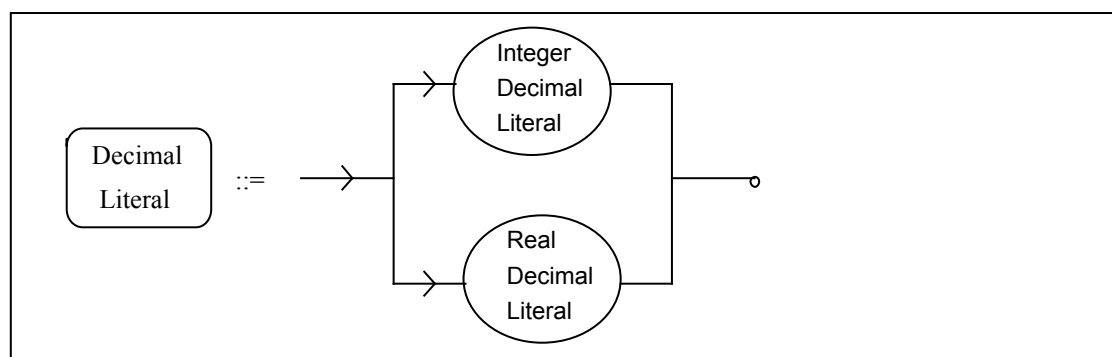


Figure 3-2: Decimal Literal Definition Diagram

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

where Integer Decimal Literal and Real Decimal Literal are defined as in figures 3-3 and 3-4:

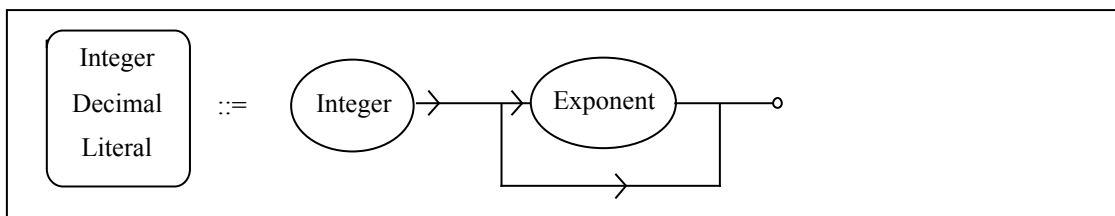


Figure 3-3: Integer Decimal Literal Definition Diagram

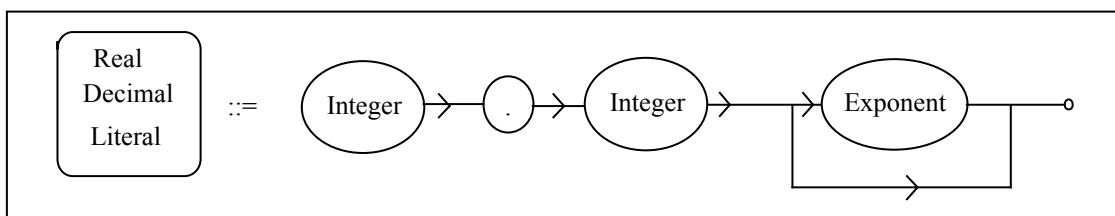


Figure 3-4: Real Decimal Literal Definition Diagram

where Integer and Exponent are defined as in figures 3-5 and 3-6:

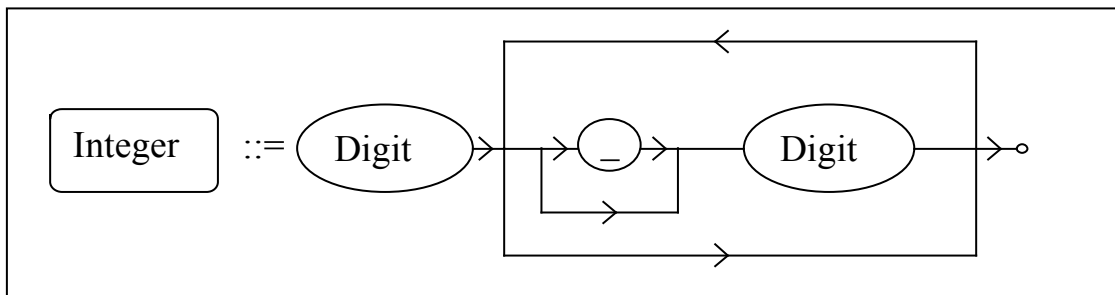


Figure 3-5: Integer Definition Diagram

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

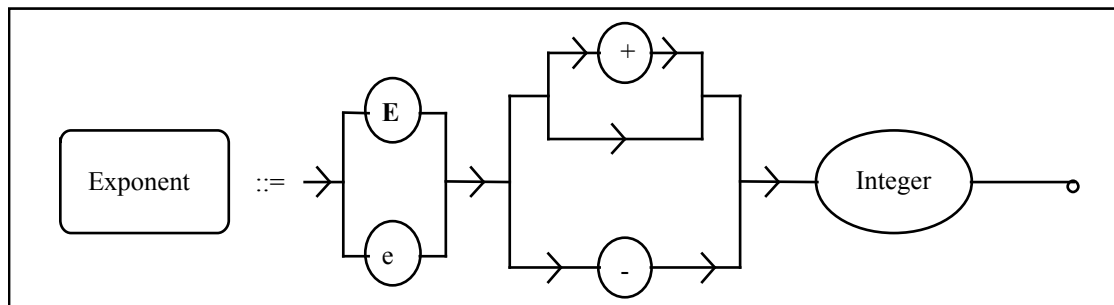


Figure 3-6: Exponent Definition Diagram

An underline character inserted between adjacent digits of a decimal literal does not affect the value of this decimal literal. The letter E of the exponent, if any, can be written either in lowercase or in uppercase, with the same meaning. Leading zeros are allowed. No space is allowed in a decimal literal.

12	0	1E6	123_456	-- integer literals
12.0	0.0	0.456	3.14159_26	-- real literals
1.3E-12	1.0E+6			-- real literals with exponent

Example 3-1: Decimal Literals

b) based literals

See figure 3-7.

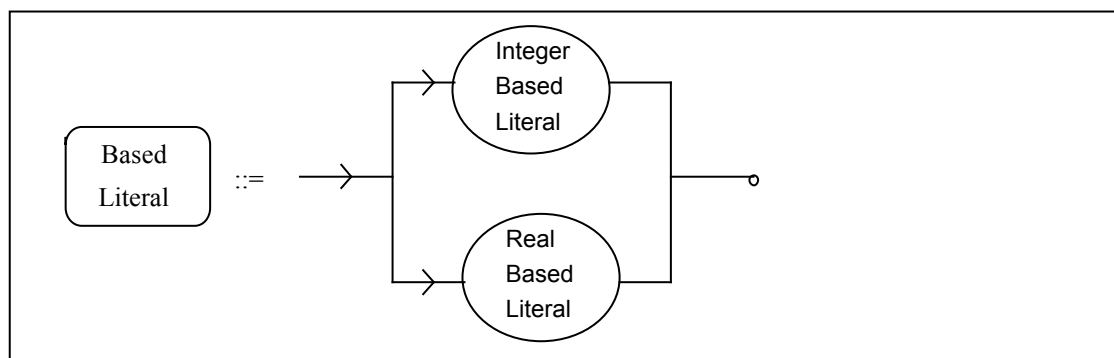


Figure 3-7: Based Literal Definition Diagram

where Integer Based Literal and Real Based Literal are defined as in figure 3-8 and figure 3-9:

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

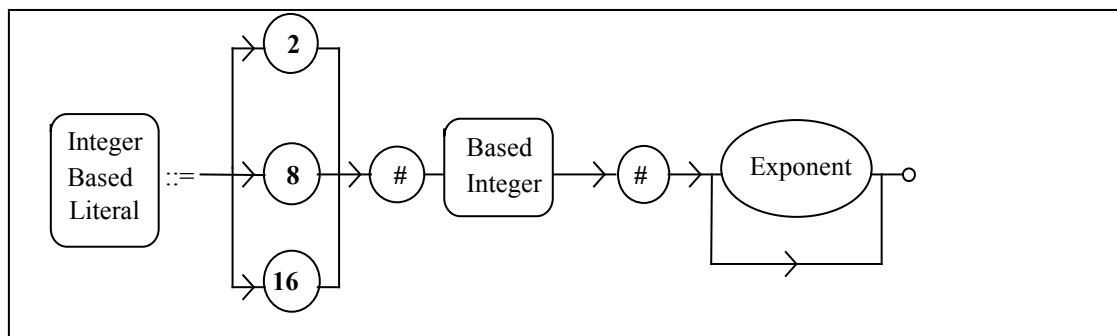


Figure 3-8: Integer Based Literal Definition Diagram

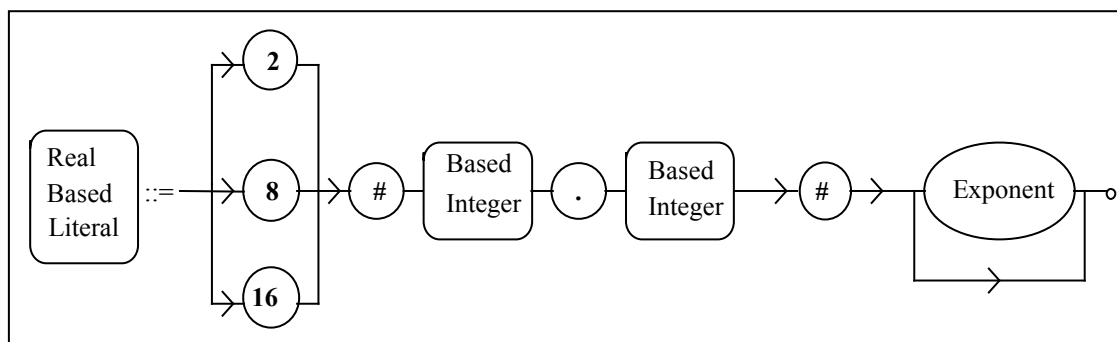


Figure 3-9: Real Based Literal Definition Diagram

where Based Integer is defined as in figure 3-10:

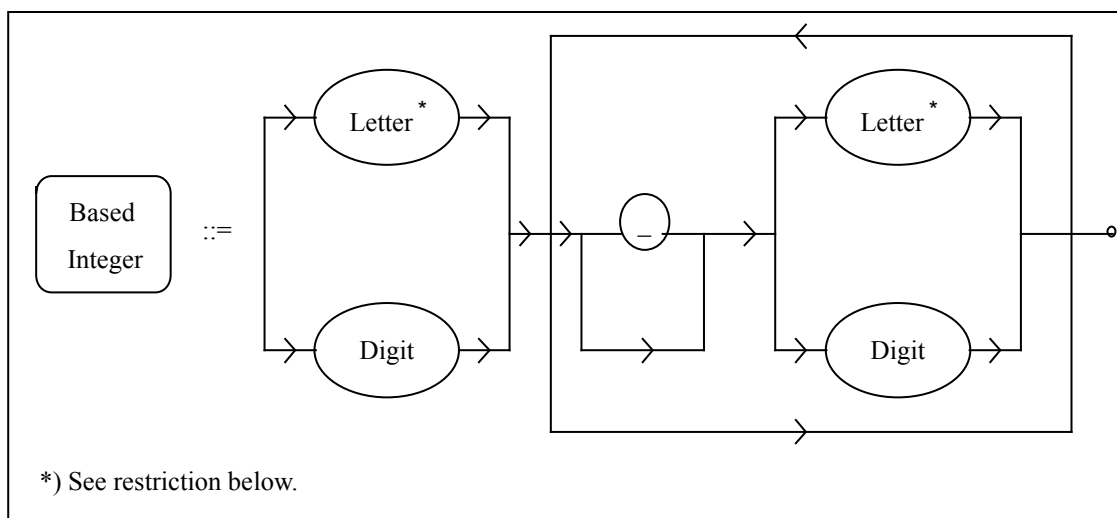


Figure 3-10: Based Integer Definition Diagram

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

The only letters allowed as extended digits are the letters A through F representing ten through fifteen. Letters are allowed for a based integer only if the base of the literal of which it is a part is 16. A letter in a based literal can be written either in lowercase or in uppercase, with the same meaning. No space is allowed in a based literal.

2#1111_1111#	16#FF#	016#0FF#	-- integer literals of value 255
16#E#E1	2#1110_0000#		-- integer literals of value 224
16#F.FF#E+2	2#1.1111_1111_111#E11		-- real literals of value 4095.0

Example 3-2: Based Literals

c) integer literals

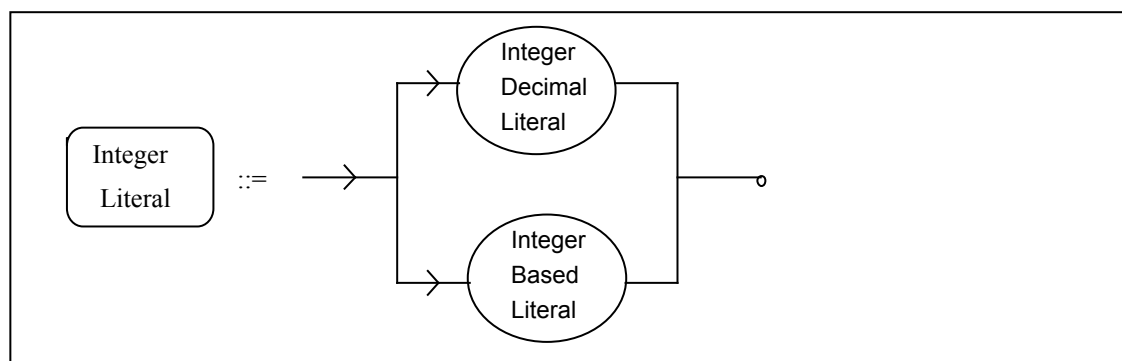


Figure 3-11: Integer Literal Definition Diagram

c) real literals

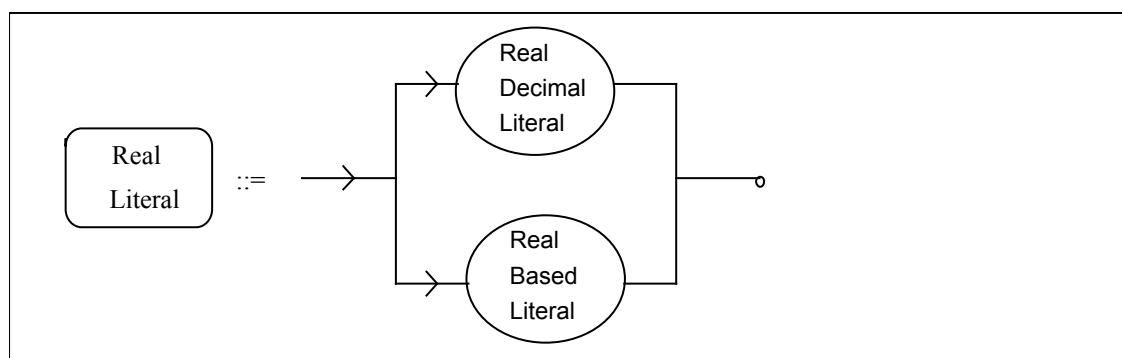


Figure 3-12: Real Literal Definition Diagram

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

3.2 LOGICAL DESCRIPTION

The logical part of an EAST DDR is composed of:

- the logical description of the models of data (using type and subtype declarations for the syntactic definition of the data, and using representation clauses for the specification of their size in bits and their location within the set of data);
- the declaration of the data occurrences, i.e., the declaration of the described data items (using object declarations).

The logical part of the Data Description Record consists of a package. This unit is introduced by the keyword **package**, followed by the package name, and ends with ‘**end package name**;’. The package name is an identifier (see 3.1.3).

The logical description package identification must be followed by the mention of the version of the EAST recommendation to which the description is supposed to conform.

As the notion of EAST recommendation version was not present in the first two EAST recommendation issues, the absence of the mention in a description should be interpreted as a reference to these two first versions (fully compatible).

A description that conforms to a particular version of the EAST recommendation must remain correct with regard to the following versions of EAST.

If an EAST description is generated using a tool, it is recommended that the tool indicate its own version using a comment.

Types are models, and objects are instances (or occurrences) of these models. Type declarations describe therefore the structure of the data elements which may occur in the described data, while **the actual data occurrences are represented by the declaration of variables and constants**.

A type (except predefined type), a subtype or a constant (except predefined constant) must be declared in the package before being used.

The declaration of variables must occur in the latter section of the logical description. Constants may be declared in the type declaration section or in the section for the declaration of variables: in the first section, they contribute to data models definition, while they represent data occurrences in the second section.

The described data is a concatenation of elements in the order of the corresponding variables. The types used in the declaration of variables must have been previously declared in the package.

Figure 3-13 summarizes the content of the logical part of a DDR.

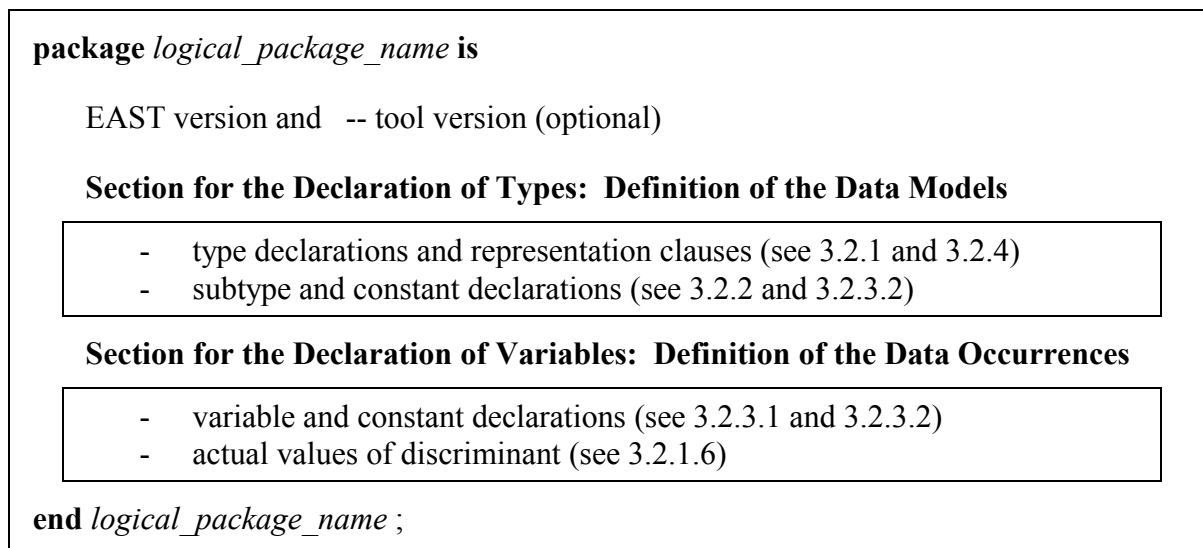


Figure 3-13: Logical Part Structure

The version declaration should respect the following format:

```
east_version : constant STRING := "3.0";
-- tool version : OASIS 5.0 (optional comment)
```

3.2.1 TYPE DECLARATIONS

The type is characterized by a set of permissible values. Several classes of types exist: scalar types (enumeration types, integer types, and real types), array types, and record types. Some types are EAST predefined types (see 3.2.1.1); the other types are user defined types and must be declared according to a specific syntax (see 3.2.1.2, 3.2.1.3, 3.2.1.4, 3.2.1.5 and 3.2.1.6).

3.2.1.1 Predefined Types

There are three predefined types provided by the EAST language: CHARACTER, STRING and EOF. Predefined means that no previous declaration has to be made explicitly by the user to use one of these types.

The predefined type CHARACTER is an enumeration type (see next subsection for the enumeration definition syntax rules), whose values are the 256 characters of the 8-bit coded Latin Alphabet No. 1 character set (see annex B and reference [1]).

The values of the predefined type STRING are one-dimensional arrays of the predefined type CHARACTER, indexed by values in increments of one of any positive integer type.

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

The number of characters must be specified every time the type is used.

As an example `STRING(1 .. 10)` designates a 10 character string, while `STRING(10 .. 22)` designates a 13 character string.

The predefined type `EOF` is exclusively used to declare a fictive end delimiter called EOF Marker (see 3.2.3.2.2).

3.2.1.2 Enumeration Type

An enumeration type is defined as a set of enumeration literals. An enumeration literal is an identifier or a character literal for one of the possible values of the type. Figure 3-14 illustrates the syntax of an enumeration type specification. Each enumeration literal yields a different enumeration value.

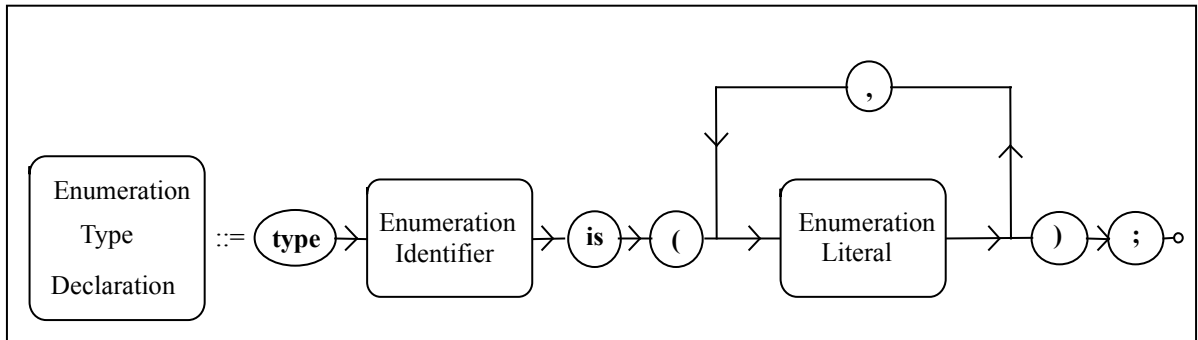


Figure 3-14: Enumeration Type Specification Diagram

where the enumeration literal is defined as in figure 3-15:

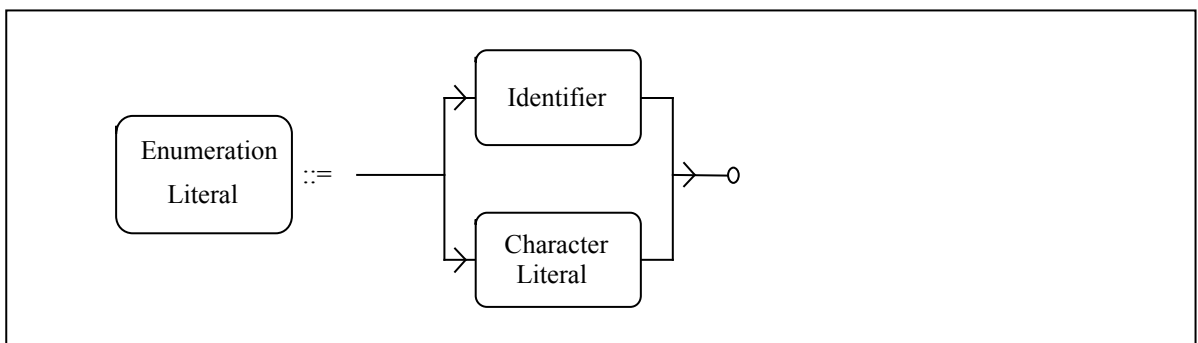


Figure 3-15: Enumeration Literal Definition Diagram

CCSDS RECOMMENDED STANDARD FOR EAST SPECIFICATION

The following example presents some enumeration type definitions.

```
type DAY is (MON, TUE, WED, THU, FRI, SAT, SUN);
type STATE is (OFF, ON);
type ROMAN_DIGIT is ('I', 'V', 'X', 'L', 'C', 'D', 'M');
```

Example 3-3: Enumeration Type Declarations

3.2.1.3 Integer Type

An integer type is defined as a set of integer values specified by a range. Each bound of the range is an integer constant identifier (see 3.2.3.2) or an integer literal (see 3.1.4). Note that both bounds need not have the same integer type and that negative bounds are allowed. The range $L \dots R$ specifies the value from L to R inclusive if the relation $L \leq R$ is true. A *null* range is a range for which the relation $R < L$ is true; no value belongs to a null range.

Figure 3-16 illustrates the syntax of an integer type specification.

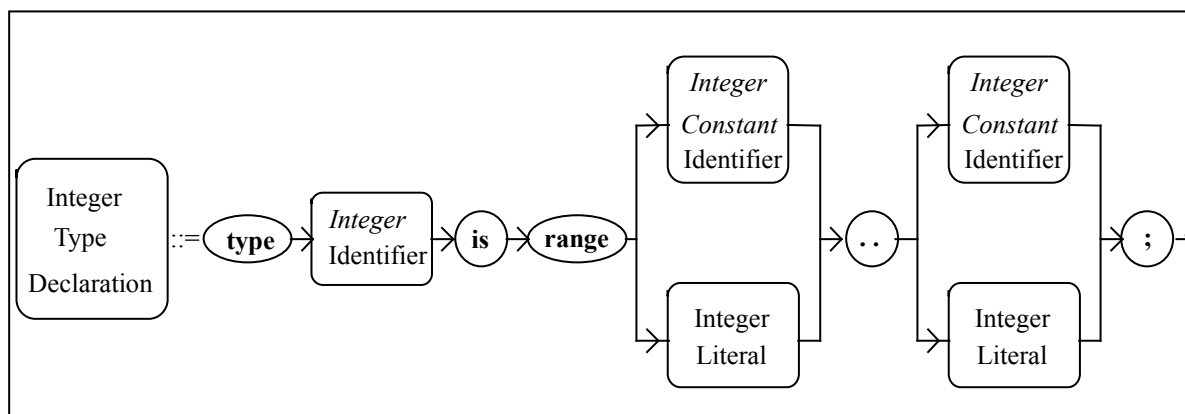


Figure 3-16: Integer Type Specification Diagram

The following example presents an integer type, defined using integer literals (-10 and 10) and an integer type, defined using a constant identifier (MAX).

```
type SMALL_INTEGER is range -10 .. 10;
type NUMBER is range 0 .. MAX;
-- where MAX could be defined as: MAX := 100;
```

Example 3-4: Integer Type Declarations