



ETSI
TECHNICAL COMMITTEE
REFERENCE TECHNICAL REPORT

TCR-TR 019

October 1994

Source: ETSI TC-SPS

Reference: DTR/SPS-02025

ICS: 33.020, 33.040.40

Key words: ASN.1

**Signalling Protocols and Switching (SPS);
Evaluation of Abstract Syntax Notation One (ASN.1) tools
for use as syntax and semantics checkers**

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE

Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE

X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

Copyright Notification: No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 1994. All rights reserved.

Contents

Foreword.....	5
Introduction.....	5
1 Scope.....	7
2 References.....	7
3 Abbreviations.....	7
4 Requirements.....	7
5 Classification.....	8
6 Contents of the test report.....	8
7 Recommended tools.....	10
7.1 CASN.....	10
7.1.1 General information.....	10
7.1.2 Tested version.....	11
7.1.3 Practical aspects.....	11
7.1.4 Test results.....	11
7.1.4.1 Unsupported ASN.1 features.....	11
7.1.4.2 Limitations of error checking.....	12
7.2 OSS ASN.1 Tools.....	12
7.2.1 General information.....	12
7.2.2 Tested version.....	13
7.2.3 Practical aspects.....	13
7.2.4 Test results.....	13
7.2.4.1 Unsupported ASN.1 features.....	13
7.2.4.2 Limitations of error checking.....	14
7.3 Siemens ASN.1 Tool Set.....	14
7.3.1 General information.....	14
7.3.2 Tested version.....	15
7.3.3 Practical aspects.....	15
7.3.4 Test results.....	15
7.3.4.1 Unsupported ASN.1 features.....	15
7.3.4.2 Limitations of error checking.....	16
8 Other tools.....	16
8.1 ASN.C.....	16
8.1.1 General information.....	16
8.1.2 Tested version.....	17
8.1.3 Test results.....	17
8.2 AsTool/T.....	17
8.2.1 General information.....	17
8.2.2 Tested version.....	18
8.2.3 Test results.....	18
8.3 CHIPSY ASN.1 Tools.....	18
8.3.1 General information.....	18
8.3.2 Tested version.....	19
8.3.3 Test results.....	19
8.4 Erlang.....	19
8.4.1 General information.....	19

	8.4.2	Tested version	19
	8.4.3	Test results.....	20
8.5		ISODE (PEPSY)	20
	8.5.1	General information	20
	8.5.2	Tested version	21
	8.5.3	Test results.....	21
8.6		OSIkit (Free Value Tool).....	21
	8.6.1	general information	21
	8.6.2	Tested version	21
	8.6.3	Test Results.....	22
8.7		MAVROS (MAVCOD).....	22
	8.7.1	General information	22
	8.7.2	Tested version	22
	8.7.3	Test results.....	23
8.8		PLC409.....	23
	8.8.1	General information	23
	8.8.2	Tested version	23
	8.8.3	Test results.....	23
8.9		Retix ASN.1 Compiler	24
	8.9.1	General information	24
	8.9.2	Tested version	24
	8.9.3	Test results.....	24
8.10		SNACC.....	25
	8.10.1	General information	25
	8.10.2	Tested version	25
	8.10.3	Test results.....	25
8.11		TWICE	25
	8.11.1	General information	25
	8.11.2	Tested version	26
	8.11.3	Test results.....	26
Annex A: Testing process			27
A.1	Test preparation.....		27
A.2	Test execution.....		27
A.3	First test report.....		27
A.4	Final test report.....		27
Annex B: Examples of test cases			28
Annex C: Table of test results			31
History			33

Foreword

This Technical Committee Reference Technical Report (TCR-TR) was prepared by the Signalling Protocols and Switching (SPS) Technical Committee of the European Telecommunications Standards Institute (ETSI).

A TCR-TR is a deliverable for use inside ETSI which records output results of ETSI Technical Committee (TC) or Sub-Technical Committee (STC) studies which are not appropriate for European Telecommunication Standard (ETS), Interim European Telecommunication Standard (I-ETS) or ETSI Technical Report (ETR) status. They can be used for guidelines, status reports, co-ordination documents, etc. They are to be used to manage studies inside ETSI and shall be mandatorially applied amongst the concerned TCs. They shall also be utilised by the TC with overall responsibility for a study area for co-ordination documents (e.g. models, reference diagrams, principles, structures of standards, framework and guideline documents) which constitute the agreed basis for several, if not all, TCs and STCs to pursue detailed standards.

Introduction

Signalling information to be exchanged between telecommunication systems becomes more and more complex. For this reason, it becomes necessary to change the description technique of signalling messages from the tabular notation to the Abstract Syntax Notation One (ASN.1).

It is considered very important that the ASN.1 which is used in standards is without errors. The same applies to ASN.1 modules which could be made available for general use.

In order to ensure that an ASN.1 specification is without errors, an automated checker could be used. A number of tools is available that could be used for this purpose. This TCR-TR presents the results of the testing of the capabilities to check ASN.1 modules for a number of these tools.

According to the test results the tools have been divided into two groups: recommended tools and other tools. The classification of the tools was based on a pre-defined set of requirements (described in clause 4), but minor discrepancies have been ignored.

The following conventions are used to refer to elements of the ASN.1 language:

- Terms which refer directly to items or productions defined by the ASN.1 specification are used between quotation marks. The terms that are defined in CCITT Recommendation X.208 (1988), § 3 are used without quotation, with the exception that sometimes "**any**" **type** is used instead of **any type** in order to prevent misunderstandings.
- Instead of **value of an integer type**, usually **integer value** is used (and similar for all other types).
- Instead of **an identifier with which an integer value is associated**, **defined integer** is used (and similar for all other types).
- Instead of "**Type**", **type** is used.
- Instead of "**DefinedType**", **defined type** is used.
- Instead of "**BuiltinType**", **built-in type** is used.
- The terms that are used in the titles of CCITT Recommendation X.208 (1988), §§ 37.1 to 37.6 (describing subtype value sets) are used instead of references to the corresponding productions (e.g. **inner subtyping** instead of "**InnerSubtyping**").

Blank page

1 Scope

The main objective of this Technical Committee Reference Technical Report (TCR-TR) is to present a list of recommended tools which can be used to ensure that an ASN.1 specification complies with the syntax and semantics defined in CCITT Recommendation X.208 [1].

In order to establish the list a number of tools have been evaluated. The evaluation was based on the list of requirements given in clause 4.

The evaluation concerns the syntax and semantics checking capabilities of the tools. Other functions, such as code generation, have not been evaluated.

It is not the objective of this TCR-TR to identify a single tool to be used by all editors and by the ETSI secretariat.

Inclusion of a tool in the list of recommended tools does not imply any guarantee for other versions of the tool or for the quality of support supplied by the provider of the tool.

2 References

For the purposes of this TCR-TR, the following references apply:

- [1] CCITT Recommendation X.208 (1988): "Specification of abstract syntax notation one (ASN.1)" (also published as ISO/IEC 8824:1990).
- [2] CCITT Recommendation X.209 (1988): "Specification of basic encoding rules for abstract syntax notation one (ASN.1)" (also published as ISO/IEC 8825:1990).
- [3] ETR 060 (1992): "Signalling Protocols and Switching (SPS); Guidelines for using Abstract Syntax Notation One (ASN.1) in telecommunication application protocols".
- [4] ITU-T Recommendation Q.773 (1993): "Specifications of Signalling System No.7; Transaction Capabilities Application Part (TCAP); Transaction capabilities formats and encoding".
- [5] ISO/IEC DIS 8824-1 (1992): "Information technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1)".
- [6] CCITT Recommendation X.219 (1988): "Remote operations: Model, notation and service definition".

3 Abbreviations

For the purposes of this TCR-TR, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules (as defined in CCITT Recommendation X.209 [2])
ROSE	Remote Operation Service Element
SNMP	Simple Network Management Protocol
TCAP	Transaction Capabilities Application Part

4 Requirements

The tool shall accept any valid ASN.1 specification (CCITT Recommendation X.208 [1]), apart from the exceptions listed below. A transformation of the ASN.1 specification by the user of the tool shall not be required (e.g. addition of semicolon between assignments, sorting of type definitions, etc.).

The tool shall indicate if a rule defined in CCITT Recommendation X.208 [1] was violated by an ASN.1 specification.

Support of contained subtyping and inner subtyping is not required.

Support of the MACRO notation is not required, as long as the usage of the ERROR or OPERATION macro (as specified in CCITT Recommendations X.219 [6] and Q.773 [4]) are supported.

Any of the following notations for choice values will be admitted:

- 1) identifier value (standard notation)
- 2) {identifier value} ({}-notation)
- 3) identifier : value (:-notation)

Any of the following notations for "any" values will be admitted:

- 1) type value
- 2) {type value}
- 3) {type : value}

NOTE: The first option is according to CCITT Recommendation X.208 [1]. Since this notation is problematic for automatic parsers, the second option has been used by some tool developers. The third option is according to the notation as specified in ISO/IEC DIS 8824-1 [5].

5 Classification

In total 10 tools were tested. The test method is described in annex A. The resulting test reports and the requirements were used as the basis for classification. Minor discrepancies between the requirements and the test results have been ignored. These limitations can be found in the individual test report for the tool.

For two tools, ISODE and OSKit, the test procedure was only partly executed. It became apparent in an early stage that the tools would not qualify as recommended tools, and therefore no detailed test report was written.

Two other tools were not tested. The provider of one of these, AsTool/T, considered that the tool would not qualify, and decided not to supply an evaluation copy. The other tool, TWICE, was not tested since the documentation made clear that it would not qualify.

The four tools that were not (fully) tested do feature in the list of tools. An indication is given of the reasons why they do not qualify as recommended tools.

NOTE 1: Both for the definition of the requirements and for the individual decisions whether a discrepancy could be considered of minor importance, the current practice of specification within SPS has been taken as a basis. These decisions may not be appropriate for different ways of specification. Notably the use of macros other than OPERATION and ERROR may cause some of the recommended tools to be of limited use. The test report lists all limitations that were found for each tool.

NOTE 2: Most of the tools do not have the checking of ASN.1 as their primary function. If such a tool is not included in the list of recommended tools, this does not imply that this tool is not suited for its primary task.

6 Contents of the test report

The test report for the recommended tools covers all items that are listed below. For the other tools, the item "practical aspects" is omitted.

- **General information**

This information was supplied by the provider of the tool. It is not a result of the evaluation.

NOTE 1: Since the ASN.1 specification is likely to change from the one given in CCITT Recommendation X.208 [1] to the one of ISO/IEC DIS 8824-1 [5] (the "1992" standard), it is indicated which of the features in this new standard are supported by the tool. These features were not tested, apart from the :-notation for choice values. Furthermore, it is indicated whether the provider plans to adapt the tool to the new standard.

- **Tested version**

- **Practical aspects**

Some of the aspects that are of importance for the practical use of the tool are described here. This information has no influence on the classification of the tools, but it may influence the ease with which the tool may be used.

- **Error reporting and recovery**

Error messages that do not indicate the type of error or the location of the error in the specification can pose problems with its correction. This type of error often occurs if a syntactical construct is used that is not supported by the tool.

Good error recovery will enable the user to find several errors in a module with one pass of the tool. This can be especially useful if the person who checks the specification is not the same as the one who is responsible for corrections, e.g. in the case where an editor uses the tool.

- **Supported notation for choice and any values**

If the tool supports another method to specify choice and any values than the one which is used in the specification that has to be tested, this means that the specification may have to be adapted.

- **Restrictions on files**

For some tool the name of the file which contains the specification has to fulfil certain requirements. Furthermore, some tools only allow one module per file.

- **Import/export**

Specifications often consist of several interdependent modules. In order to check such specifications, these modules will have to be organised according to the scheme that the used tool supports. This may involve putting all modules in separate files with specific file names and/or sorting the modules according to dependencies.

In case there are circular dependencies between modules, it is not possible to fulfil the requirements that are placed on the organisation of the input by some of the tools.

- **Test results**

For the recommended tools, a full report is given of all errors that where found. For the other tools, a list of the most serious limitations is given. In some cases, reference is made to examples which are given in annex B.

If a limitation corresponds to usage of ASN.1 which is deprecated in ETR 060 [3] (and which, therefore, should not occur), this is indicated.

NOTE 2: A concise overview of the test results for all tested tools can be found in annex C.

NOTE 3: The number of items in the list of limitations is not significant. If a tool supports a certain feature, but there are some limitations in this support, this will cause more items to appear in the list than the case of not supporting the feature at all.

The test results are divided in two groups:

- **Unsupported ASN.1 features**

These limitations will generally make it impossible to parse a specification in which such a feature occurs.

NOTE 4: There are some syntactical constructs that are very difficult to parse: the definition and use of macros and the (standard) value notation for choice and any types. None of the tools fully supports all of these items.

- **Limitations of error checking**

These are mostly missing error messages for semantic errors. These limitations do not prohibit the parsing of a module. However, the usefulness of a tool decreases with the increasing number of errors that are not detected by it.

NOTE 5: CCITT Recommendation X.208 [1] specifies some error conditions which are very difficult to check. Examples of this are the requirement that the subtype notation shall not be used so as to produce a subtype with no values and the requirement that it should be possible to define values which have a finite representation for a recursive type.

7 Recommended tools

Using the method of classification described in clause 5, the tools that are listed in this clause have been classified as recommended tools. They are listed in alphabetical order.

7.1 CASN

7.1.1 General information

Originating company:

NOKIA Research Center

P.O. Box 45

FIN-00211 Helsinki

FINLAND

Phone: +358 0 43 761

Fax: +358 0 43 76227

Contact point:

Ari Ahtiainen

E-mail: aanen@research.nokia.com

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the Basic Encoding Rules (BER).
Target languages: C. Support for C++ will be available in 1994.
- Generation of interactive test functions.

Platform(s) on which the tool can be executed:

UNIX, VMS, MS-DOS.

Supported 1992 features:

The new CASN 2.01 syntax-checker (beta test version available) accepts the following "1992 notation" as specified in ISO/IEC DIS 8824-1 [5]:

- extended character string types;
- information object specification;
- constraint specification;
- parametrization of ASN.1 specifications.

Schedule for the implementation of the 1992 notation:

See above.

Support:

Correction of documented reproducible errors appearing during the first 2 months.
Maintenance contracts are offered to cover error-corrections and/or new features in the software.

Price:

with C source code of Run Time Support system:
MS-DOS: US\$ 10 500,- UNIX: US\$ 21 000,- VMS: US\$ 27 000,-
without C source code of Run Time Support system:
MS-DOS: US\$ 7 500,- UNIX: US\$ 15 000,- VMS: US\$ 21 500,-
only syntax analysis part:
MS-DOS: US\$ 2 500,- UNIX: US\$ 4 000,- VMS: US\$ 8 000,-

7.1.2 Tested version

v1.54 for MS-DOS Version 5.0.

7.1.3 Practical aspects

Notation for choice and any values:

The {}-notation must be used for choice values and any values¹⁾.

Restrictions on files:

There are no restrictions on file names.
Not more than one module can be in a file.

Import/export:

If a module imports symbols from other modules, these modules have to be compiled first. During compilation a file is generated which contains information concerning the symbols that are defined in the module. This file is used for the necessary information if another module imports one of those symbols.

Error reporting and recovery:

The tool generates a file containing an error listing. For most errors there is an appropriate description in the context of the ASN.1 source file. However, for some syntax errors messages were generated without indication of the line number or the type of error.
There is some error recovery from syntax errors.

7.1.4 Test results

7.1.4.1 Unsupported ASN.1 features

- a) Macro definition is not supported. The use of the OPERATION and ERROR macros is possible²⁾, with the following limitations:
 - the resulting value of a choice type has to be replaced by an integer value;
 - macro type notation may only appear as a type in type or value assignments, but not as element of set or sequence type;
 - built-in types cannot be used in the type notation.
- b) Inner subtyping is not supported.
- c) Contained subtyping is not supported.
- d) The selection type is not supported.
- e) The abbreviations "SET" and "SEQUENCE" for "SET OF ANY" and "SEQUENCE OF ANY" are not supported.
- f) Named values without identifiers are not supported.

NOTE: Use of such values is deprecated in ETR 060 [3].

¹⁾ Version 2 will support the :-notation. A beta release of this version is available.

²⁾ According to the provider of the tool the BIND and UNBIND macros as defined in CCITT Recommendation X.219 [6] are also supported. These were not tested.

- g) Importation of symbols from different modules with the same module reference is not supported (see example 8).
- h) The value "2" as base for real values is not accepted.
- i) Non-printable ASCII characters cannot be used in character string types.
- j) The value notation for the external type is not accepted.
- k) Specification of alternatives for subtypes (using "|") is not supported except for the use within a permitted alphabet subtype specification.
- l) The value "0" cannot be used as upper endpoint of a size constraint subtype.

7.1.4.2 Limitations of error checking

- a) The content of string values is not checked.
- b) Subtypes can be defined in a way that no values can be specified (see example 5 and also note 5 in clause 6).
- c) Recursive type definitions for which no values can be specified are not detected (see example 3 and also note 5 in clause 6).

7.2 OSS ASN.1 Tools

7.2.1 General information

Originating company:

Open Systems Solutions, Inc.
301 North Harrison Street, Suite 265
Princeton
New Jersey 08540
USA

Contact point:

E-Mail: info@oss.com

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the BER.
Target languages: C, Pascal.

Platform(s) on which the tool can be executed:

UNIX, VMS, OS/2, MS-DOS, MVS, Apple Macintosh, pSOS+, Windows NT, Novell Netware, Tandem Guardian, NeXT, Transputer.

Supported 1992 features:

- :-notation for any values and choice values;
- optionally the following ASN.1 features can be parsed/checked:
 - the improved subtype notation;
 - restrictions on the specification as given in CCITT Recommendation X.208 [1] (no named types without identifiers, no defined value in object identifiers that are used in the definition of an ASN.1 module);
 - user-defined constraints;
 - information objects;
 - table constraints.

Schedule for the implementation of the 1992 notation:

The following will be available in the 4th quarter of 1993:

- component relation constraints;
- parametrization of ASN.1 specifications.

Support:

24 hours/day customer support and maintenance, including new releases and features, for the first year are included in the license prices.

7.2.4.2 Limitations of error checking

- a) Implicit tagging of choice types is permitted⁶⁾.
- b) Implicit tagging of "any" types is permitted⁷⁾.
- c) Values that are equal to the open endpoints of a value range subtype are accepted.
- d) Subtypes can be defined in a way that no values can be specified (see example 5 and also note 5 in clause 6).
- e) Recursive type definitions for which no values can be specified are not detected (see example 3 and also note 5 in clause 6).
- f) Circular production rules can be used within a macro definition.
- g) A named value without identifier can be assigned to a choice type which has more than one alternative for which the value is valid, making the value ambiguous.

NOTE: The use of named values without identifiers is deprecated by ETR 060 [3].

- h) "MAX" and "MIN" are not interpreted in the right way. The values of the endpoints of the underlying built-in types are used, instead of the endpoints of the parent type.

7.3 Siemens ASN.1 Tool Set

7.3.1 General information

Originating company:
Siemens Austria
Autokaderstraße 29
A-1210 Vienna
AUSTRIA

Contact point:
Heinz Schaefer
Phone: +43 1 27705 381
Fax: +43 1 27705 210

Other relevant functions (apart from checking ASN.1 specifications):

- generation of cross reference information;
- expansion of defined types to the constituent basic types;
- interactive encoding and decoding, using the BER.

Platform(s) on which the tool can be executed:
MS-DOS.

Supported 1992 features:
Warnings are issued at the use of named types without identifier.

Schedule for the implementation of the 1992 notation:
There is no fixed schedule. The 1992 notation will be implemented as needed for internal use by Siemens.

Support:
There is no support contract available. According to the provider reported errors will be corrected immediately, or in the next release. Releases usually take place four times a year.

5) According to the provider of the tool this problem is related to the use of third party security software in the evaluation copy of the software. This security software is not used for production versions.
6) This problem has been solved in a new release.
7) This problem has been solved in a new release.

Price:

Syntax checker only:

AS 25 000,-

Syntax checker with additional information generation functions:

AS 40 000,-

Syntax checker with additional information generation functions and encoder and decoder:

AS 60 000,-

7.3.2 Tested version

P1.7A for MS-DOS 5.0.

7.3.3 Practical aspects

Notation for choice and any values:

The standard notation can be used for choice values. Value assignment for "any" values is not supported.

Restrictions on files:

The filenames have to be the same as the name of the module that they contain. If the module name is longer than eight characters, the filename has to be the same as the first eight characters of the module name. This causes problems if there are several modules with names which do not differ in these first eight characters.

The filenames have to end in a suffix of ".asn".

It is not possible to have more than one module in a file.

Import/export:

If a module imports symbols, the exporting module is automatically compiled (the file name convention is used to find the appropriate file). A search path can be specified on the command line if the file is not in the current directory.

Error reporting and recovery:

Error messages are written to a file. In this file the location of the error is indicated and for semantic errors there is some information on the type of error.

There is some error recovery from syntax errors.

7.3.4 Test results

7.3.4.1 Unsupported ASN.1 features

- a) Macro definition is not supported. The use of the OPERATION and ERROR macros is supported. The recognised macros can only be used if they are imported. Since the file from which they are imported is also compiled, there are error messages for the macro definition. The following restrictions were found for the use of the macro:
 - 1) it is not possible to use a defined value in the value notation;
 - 2) it is not possible to use a built-in type after the "ERRORS" or "LINKED" keyword of the OPERATION macro.
- b) The abbreviations "SET" and "SEQUENCE" for "SET OF ANY" and "SEQUENCE OF ANY" are not supported.
- c) Value assignment for "any" types is not supported.⁸⁾
- d) Defined values in a named bit list, enumerated type or named number list are not supported.
- e) The use of named values without identifier is not supported.

NOTE: The use of named values without identifiers is deprecated by ETR 060 [3].

⁸⁾ According to the provider of the tool the future release 1.11A will be able to handle values for "any" types.

- f) The selection type is not supported.
- g) Values for choice types are not accepted if the identifier that indicates the selected alternative is followed by a defined value or by another identifier (e.g. an identifier from an enumerated type). This does not apply if the choice type is an element of a structured type (see note 4 in clause 6).
- h) Inner subtyping is not supported.
- i) Contained subtyping is not supported.
- j) The single value subtype is only supported for the integer type.
- k) Value range subtype is only supported for the integer type.
- l) Open endpoints for a value range subtype are not accepted.
- m) "MIN" and "MAX" as endpoints of a value range subtype are not accepted.
- n) "Externaltypereference" and "Externalvaluereference" are not supported.

NOTE: Use of "Externaltypereference" and "Externalvaluereference" is deprecated by ETR 060 [3].

- o) Recursive type definitions are not supported.⁹⁾
- p) Importation from different modules with the same module reference is not possible (see example 8).
- q) Usage of a real type as element of a sequence type causes the program to terminate with an internal error.

7.3.4.2 Limitations of error checking

- a) The contents of character string values (IA5String, Generalized Time, etc.) are not checked.
- b) The numbers and identifiers in a named bit list, enumerated type and named number list are not checked for uniqueness.
- c) There is no check for duplicate tags in a set type.
- d) It is possible to specify "any" types with implicit tags.
- e) It is possible to specify choice types with implicit tags (only if "IMPLICIT TAGS" is the default).
- f) The same "APPLICATION" class tag can be assigned twice in the same module.
- g) There is no check whether a subtype is appropriate for the parent type, if this parent type is a defined type.
- h) A defined integer with a negative value that is used as a tag causes a warning that tags greater than 30 cannot be evaluated for distinctness.
- i) Subtypes can be defined in a way that no values can be specified (see note 5 in clause 6).

8 Other tools

Using the method of classification described in clause 5, the tools that are listed in this clause have not been classified as recommended tools. They are listed in alphabetical order.

8.1 ASN.C

8.1.1 General information

Originating company:

DSET Corporation
440D County Road 513
Califon
New Jersey 07830
USA
Phone: +1 908 832 6533
Fax: +1 908 832 6523

Contact point:

Alex Kuo
E-Mail: dset!alex@uunet.UU.NET

⁹⁾ According to the provider of the tool release 1.9A is able to handle types which are directly recursive.

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the BER or XDR.
Target languages: C and C++.

Platform(s) on which the tool can be executed:

UNIX, pSOS+ (an earlier version exists for MS-DOS).

Supported 1992 features:

None.

Schedule for the implementation of the 1992 notation:

The implementation of the 1992 standard is planned for 1994.

Support:

Standard technical support procedures apply.

Price:

Developer's license starts at US\$ 9 900,-

8.1.2 Tested version

v3.2 for Sun - SPARC with SunOs 4.1.3.

8.1.3 Test results

The limitations that were found include:

- a) The macro notation is not supported (X.400, X.500, Transaction Capabilities Application Part (TCAP), Simple Network Management Protocol (SNMP) and Remote Operation Service Element (ROSE) macros should be supported according to the documentation, but compilation failed).¹⁰⁾
- b) It is not checked whether values for subtypes satisfy the specified constraints.
- c) Several tagging rules are not checked.

8.2 AsTool/T

8.2.1 General information

Originating company:

KDD R&D
Bldg 2-1-23 Nakameguro
Meguroku
Tokyo 153
JAPAN

Contact point:

Mr Hasegawa
Phone: +81 492 66 7368
Fax: +81 492 66 7510
E-Mail: hasegawa@osi.kddlab.kddlabs.co.jp

Other relevant functions (apart from checking ASN.1 specifications):

AsTool/T is a tool for testing OSI application protocol software. It consists of a pre-processor, ASN.1 encoder and decoder, and a user interface.

Platform(s) on which the tool can be executed:

UNIX, VMS, MS-DOS.

¹⁰⁾ According to the tool provider these macros are normally supported, but they were left out of the evaluation copy.

Supported 1992 features:

Unknown.

Schedule for the implementation of the 1992 notation:

Unknown.

Support:

Unknown.

Price:

US\$ 20 000,-

8.2.2 Tested version

The tool was not tested.

8.2.3 Test results

The provider of the tool decided not to supply an evaluation copy, since the tool was not designed as a syntax and semantics checker. It does not support the value notation and module definitions.

8.3 CHIPSY ASN.1 Tools

8.3.1 General information

Originating company:

KVATRO A/S
Pirsenteret
N-7005 Trondheim
NORWAY

Contact point:

Bernt Marius Johnsen
Phone: +47 7 52 00 90
Fax: +47 7 52 01 40
E-Mail: bernt@kvatro.no

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the BER.
Target language: CHILL.
- Syntax directed editing (using the Emacs editor).

Platform(s) on which the tool can be executed:

UNIX, VAX/VMS

Supported 1992 features:

:-notation for specifying values for choice and any values.

Schedule for the implementation of the 1992 notation:

Unknown.

Support:

Maintenance agreements which include telephone or on-line support are available. Special support, including long term agreements, is also available.

Price:

For UNIX workstations: NOK 37 500,-

8.3.2 Tested version

v12 for Sun - SPARC with SunOs 4.1.3.

8.3.3 Test results

The limitations that were found include:

- a) Macros are not supported.
- b) The import and export mechanism did not work (according to the documentation such a mechanism does exist).
- c) It is not checked whether values for subtypes satisfy the specified constraints.
- d) No semantic checking is performed on the definition of subtypes (e.g. it is possible to use contained subtypes with an integer parent type).

8.4 Erlang

8.4.1 General information

Originating company:
ELLEMTEL Utvecklings AB
P.O. Box 1505
S-125 25 Älvsjö
SWEDEN

Contact point:
Claes Wikström
Phone: +46 8 727 39 83
Fax: +46 8 647 82 76
E-mail: klacke@erix.ericsson.se

Other relevant functions (apart from checking ASN.1 specifications):

Erlang is primarily a programming language designed for real-time applications. With the language an ASN.1 to Erlang compiler is delivered. This compiler (itself written in Erlang) generates BER encoders and decoders in Erlang.

Platform(s) on which the tool can be executed:

- UNIX.
- WxWorks for FORCE card.

Supported 1992 features:

None.

Schedule for the implementation of the 1992 notation:

Unknown.

Support:

Unknown (see below).

Price:

The current version of Erlang is not commercially available. Release 4.1 will be sold as a product. For non-commercial use it may be possible to get a free copy of the current version.

8.4.2 Tested version

Version 2.2 for Sun - SPARC with SunOs 4.1.3.

8.4.3 Test results

The limitations that were found include:

- a) Macros are not supported.
- b) The real type is not supported.
- c) The accepted syntax for import and export does not conform to CCITT Recommendation X.208 [1].
- d) Values are not checked, neither on type nor on conformance to subtype constraints.

8.5 ISODE (PEPSY)

8.5.1 General information

Originating company:

ISODE was developed by a large number of people. At the moment, the ISODE Consortium works on further developing the ISODE OSI tool set.

ISODE Consortium
European Office
P.O. Box 505
London SW11 1DX
UK
Phone: +44 71 223 4062
Fax: +44 71 223 3846
E-Mail: ic-info@isode.com

X-Tel Services Ltd. provide support for the ISODE packages on a commercial basis. X-Tel will also enhance and market ISODE Consortium products.

X-Tel Services Ltd.
University Park
Nottingham NG7 2RD
UK
Phone: +44 602 412648
Fax: +44 602 790278
E-Mail: support@xtel.co.uk

Contact point:
None.

Other relevant functions (apart from checking ASN.1 specifications):

The PEPSY ASN.1 compiler is only one out of a number of modules. All modules are related to the development of OSI applications and protocols. The PEPSY compiler generates encoders and decoders for the BER.

Target language: C.

Platform(s) on which the tool can be executed:
UNIX.

Supported 1992 features:
None.

Schedule for the implementation of the 1992 notation:
Unknown.

Support:
Support is available from X-Tel Services Ltd (see above).

Price:

Version 8.0 is free. It can be downloaded from uu.psi.com. Newer versions may be available from X-Tel Services Ltd.

8.5.2 Tested version

Version 8.0.

8.5.3 Test results

The limitations that were found include:

- a) No checks are performed on values.
- b) No checks are performed on the definition of subtypes.
- c) No checks are performed on tagging.
- d) Macros are not supported.

8.6 OSKit (Free Value Tool)

8.6.1 general information

Originating company:

NIST
The Formal Methods Project
Nat. Inst. of Standards and Technology
Bldg. 225, Room B141
Gaithersburg, MD 20899
USA
E-mail: estelle@osi.ncsl.nist.gov

Contact point:

None.

Other relevant functions (apart from checking ASN.1 specifications):

OSKit is a collection of tools for the application of Estelle and ASN.1. The Free Value Tool forms part of this.

The Free Value Tool generates a (non-executable) representation of the parsed ASN.1. This could be used as input for a back-end.

Platform(s) on which the tool can be executed:

UNIX.

Supported 1992 features:

None.

Schedule for the implementation of the 1992 notation:

Unknown.

Support:

None.

Price:

Free.

8.6.2 Tested version

Not specified.

8.6.3 Test Results

The limitations that were found include:

- a) No checks are performed on values.
- b) The "real" type is not supported.
- c) The enumerated type is not supported.
- d) Subtypes are not supported.
- e) No checks are performed on tagging.
- f) Macros are not supported.

8.7 MAVROS (MAVCOD)

8.7.1 General information

Originating company:

The tool is owned by INRIA and SNI (joint ownership). It is sold by BULL and E3X (licensees of INRIA).

INRIA
2004, route des Lucioles
F-06902 Sophia Antipolis
FRANCE
Phone: +33 93 65 77 77
Fax: +33 93 65 77 65

Contact point:

Christian Huitema
E-Mail: huitema@mitsou.inria.fr

Other relevant functions (apart from checking ASN.1 specifications):

MAVCOD is the first pass of the MAVROS ASN.1 compiler. This compiler generates decoder and encoder code for the BER.
Target languages: C.

Platform(s) on which the tool can be executed:

UNIX, OS/2, Apple Macintosh, MS-DOS.

Supported 1992 features:

- :-notation for choice values.

Schedule for the implementation of the 1992 notation:

It is planned that ISO/IEC DIS 8824-1 [5] shall be supported in the future. No fixed schedule is available.

Support:

INRIA does not officially provide support. For a number of years, releases of up-to-date versions will be provided every 6 months. Support may be available from BULL or E3X.

Price:

(INRIA sold the compiler (including code generator) for FF 40 000,- for a full source release).

8.7.2 Tested version

v2.0 for Sun - SPARC with SunOs 4.1.3.

8.7.3 Test results

The limitations that were found include:

- a) Subtype constraints are not checked.

8.8 PLC409

8.8.1 General information

Originating company:

Marben Produits
11, rue Curie
F-92150 Suresnes
FRANCE
Phone: +33 1 45 06 32 64
Fax: +33 1 47 72 55 00

Contact point:

M. Jamal
Phone: +33 1 45 06 32 31
Fax: +33 1 45 01 23 31

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the BER.
Target languages: C, PASCAL and COBOL.

Platform(s) on which the tool can be executed:

UNIX.

Supported 1992 features:

None.

Schedule for the implementation of the 1992 notation:

Unknown.

Support:

The license fee provides a six months warranty against defects.

Price:

Unknown.

8.8.2 Tested version

v5.0B for Sun - SPARC with SunOs 4.1.3

8.8.3 Test results

The limitations that were found include:

- a) Modules can only be parsed if all definitions are separated by semicolons.
- b) Macro definitions are supported, but with the restriction that the OPERATION macro as specified in CCITT Recommendation X.219 [6] cannot be compiled. Furthermore, it is impossible to specify a macro which could replace the OPERATION macro in such a way that every possible instance of the original would also be a valid instance of the replacement.
- c) It is not checked whether values for subtypes satisfy the specified constraints.

8.9 Retix ASN.1 Compiler

8.9.1 General information

Originating company:

Retix
2401 Colorado Avenue
Santa Monica
California 90404-3563
USA

Contact point:

Jeff Marx
Tel: +1 310 828 34 00
Fax: +1 310 828 22 55

Contact in France:

Darragh Stokes
Tel: +33 1 30 70 60 60
Fax: +33 1 30 70 67 12
X.400: C=fr; A=Atlas; P=Retix

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the BER.
Target language: C.

Platform(s) on which the tool can be executed:

UNIX.

Supported 1992 features:

None.

Schedule for the implementation of the 1992 notation:

Full support for the 1992 notation is currently planned, and is being scheduled according to customer requirements.

Support:

There are three levels of support offered:

Phone Support	10% of license fee per annum.
Phone & Update Service	17% of license fee per annum.
High Priority Phone & Update Service	20% of license fee per annum.

Price:

Sun:	US\$ 8 965,-
PC:	US\$ 5 195,-
VAX	US\$ 12 995,-

8.9.2 Tested version

v2.09 for Sun - SPARC with SunOs 4.1.3.

8.9.3 Test results

The limitations that were found include:

- A Retix specific notation has to be used for the value notation for structured types (sequence, set, sequence of, set of, choice, any).
- The macro notation is not supported.

8.10 SNACC

8.10.1 General information

Originating company:

SNACC is in the public domain. It was developed at:

Department of Computer Science
University of British Columbia
6356 Agricultural Rd.
Vancouver, British Columbia
CANADA, V6T 1Z2

Contact point:

None.

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the BER.
Target languages: C and C++.

Platform(s) on which the tool can be executed:

UNIX.

Supported 1992 features:

None.

Schedule for the implementation of the 1992 notation:

Unknown.

Support:

None.

Price:

Free.

8.10.2 Tested version

1.0 for Sun - SPARC with SunOs 4.1.3.

8.10.3 Test results

The limitations that were found include:

- Integer, boolean and object identifier values are processed, but there is no check for the conformance to subtype constraints. Values for other types are ignored.
- No semantic checking is performed on the definition of subtypes.

8.11 TWICE

8.11.1 General information

Originating company:

Alcatel TITN Answare
1, rue Galvani
BP 110
F-91301 Massy CEDEX
FRANCE
Phone: +33 1 69 81 11 00
Fax: +33 1 69 20 15 04

Contact point:

Claude Favreau

Other relevant functions (apart from checking ASN.1 specifications):

- Generation of decoder and encoder code for the BER.
Target language: C.

Platform(s) on which the tool can be executed:

UNIX.

Supported 1992 features:

Unknown.

Schedule for the implementation of the 1992 notation:

Unknown.

Support:

Unknown.

Price:

Unknown.

8.11.2 Tested version

The tool was not tested.

8.11.3 Test results

Since the manual made clear that the tool would not qualify as a recommended tool, it was not tested.

The limitations that are mentioned in the manual include:

- a) Definitions have to be sorted in such a way that there are either only backward references or only forward references.
- b) Only value assignments for simple types can be handled.

Annex A: Testing process

The overall testing process has been divided into four phases:

- test preparation;
- test execution;
- collection of the results in a test report; and
- processing of comments from the tool providers.

The individual steps in these phases are described in the following clauses.

A.1 Test preparation

During the test preparation the requirements for an ASN.1 tool have been extracted from the standard and test suites containing tests checking each requirement have been developed.

The steps that have been executed are:

- 1) Requirements have been extracted from CCITT Recommendation X.208 [1] and a list of test purposes has been produced that describes in an informal way the objective of every test. The test purposes were divided in two classes:
 - a) valid behaviour;
 - b) invalid behaviour:
 - syntax errors;
 - semantic errors.The valid behaviour tests verify that valid ASN.1 specifications are processed correctly by a tool. Invalid behaviour tests verify that syntax or semantic errors are detected by a tool.
- 2) Based on the list of test purposes, a list of corresponding test cases has been developed. Each test case is an ASN.1 specification that is suited to carry out the test described in the corresponding test purpose. The number of ASN.1 elements in a test case that are not relevant to the test purpose has been minimised. In total 258 valid tests and 117 invalid tests were written. Since several ways of notation for some types were considered valid, corresponding versions were written for those test cases that used these types.
- 3) Some ASN.1 specifications from existing standards have been added as realistic examples to the test suites.

A.2 Test execution

In the test execution phase the test suite was executed with every tool. A simple indication of the result of the test case (pass or fail) was entered in a database. Error messages were stored in a file.

A.3 First test report

All relevant information was put into the form of a test report for each tool. As a basis for this report the database with test results and the file with error messages was used. Additional tests were executed if necessary. Each test report was sent to the concerned tool provider, to enable them to comment on the results.

The results of the tests with existing standards were used to formulate the items in the test report that concern practical aspects of testing, such as error recovery.

A.4 Final test report

Comments from the providers concerning the tested release of the tools were processed. Comments concerning improvements in later or future releases were added as notes.

Annex B: Examples of test cases

```
-- (INVALID)
-- Use of selection type as a named type
--
Example-1 DEFINITIONS ::=
BEGIN

TypeA ::= SEQUENCE{
    selectA < ChoiceA,
    name2 INTEGER}

ChoiceA ::= CHOICE{
    selectA INTEGER,
    selectB BOOLEAN}

valueA TypeA ::= {selectA 2, name2 3}

END
```

Figure B.1: Example 1

```
-- All types can have a value:
-- valueA A ::= {}
-- valueB B ::= {{d {}}}
-- valueC C ::= {d {}}
-- valueD D ::= {}
-- valueE E ::= a {}
--
Example-2 DEFINITIONS ::=
BEGIN
A ::= SEQUENCE OF B
B ::= SET SIZE(1..12) OF C
C ::= SET {
    d    D}
D ::= SEQUENCE{
    e    E}
E ::= CHOICE {
    a    A,
    c    C}

END
```

Figure B.2: Example 2

```
-- (INVALID)
-- None of the types can have a value:
--
Example-3 DEFINITIONS ::=
BEGIN
A ::= SEQUENCE (SIZE 1) OF B

B ::= SET SIZE(1..12) OF C

C ::= SET {
    d      D}

D ::= SEQUENCE{
    e      E}

E ::= CHOICE {
    a      A,
    c      C}

END
```

Figure B.3: Example 3

```
-- The reason why some things changed in the 1992 standard
--
Example-4 DEFINITIONS ::=
BEGIN

    a B ::= c d E ::= F g H ::= I

    F ::= BOOLEAN
    g BOOLEAN ::= TRUE
    I ::= INTEGER
    c INTEGER ::= 10
    E ::= ANY
    B ::= INTEGER

END
```

Figure B.4: Example 4

```
-- (INVALID)
-- Definition of a subtype for which no values are possible.
--
Example-5 DEFINITIONS ::=
BEGIN

TypeA ::= IA5String ("toto" | "papa" | "mama")
TypeB ::= TypeA (SIZE (3))

END
```

Figure B.5: Example 5

```
-- The "string" item
--
Example-6 DEFINITIONS ::=
BEGIN

TEST MACRO ::=
BEGIN
    TYPE NOTATION      ::= string "Name"
    VALUE NOTATION     ::= value (VALUE INTEGER )
END

TypeA ::= TEST abcd Name
valueA TypeA ::= 2

END
```

Figure B.6: Example 6

```
-- The returned type of the macro depends on the instance of the
-- value notation.
--
Example-7 DEFINITIONS ::=
BEGIN

TEST MACRO ::=
BEGIN
    TYPE NOTATION ::= empty
    VALUE NOTATION ::= value (VALUE INTEGER) |
                       value (VALUE BOOLEAN)
END

TestType ::= TEST
valueA TestType ::= 2
valueB TestType ::= TRUE

END
```

Figure B.7: Example 7

```
-- Importing symbols from different modules with the same module reference
--
Example-8 DEFINITIONS ::=
BEGIN
IMPORTS      TypeA FROM A-Module {iso standard 8571 pci(1)}
             TypeB FROM A-Module {ccitt recommendation 987};

TypeC ::= TypeA
TypeD ::= TypeB

END
```

Figure B.8: Example 8

Annex C: Table of test results

Table C.1 is meant to provide a quick overview over the capabilities of the tools. Only limitations that may cause serious problems when testing an ASN.1 module and inability to find an important class of errors are listed. For a complete overview of the limitations of a tool, the reader is referred to the individual test report.

NOTE: Remarks from the tool providers concerning limitations that have been solved in later releases have not been included in this table. Such remarks may be found in the individual test reports.

Boxes that are marked with "-" indicate that the corresponding feature is not supported by the tool.

Boxes that are marked with "○" indicate that there are limitations in the implementation of the corresponding feature. In those cases where a simple description of the limitation could be given, a note is provided to indicate the nature of the limitation.

Boxes that are marked with "+" indicate that there are no serious limitations in the implementation of the corresponding feature. In some cases a note indicates a minor limitation.

In the bottom row it is indicated which type of value notation is supported for choice and any values. "S" indicates the standard notation, "{" the {}-notation, ":" the :-notation and "N" indicates that another notation (which was not evaluated) has to be used.

Table C.1

	ASN.C	CASN	ERLANG	KVATRO	MAVCOD	OSS	PLC 409	RETX	SIEMENS	SNACC	
boolean	+	+	+	+	+	+	+	+	+	+	builtin types
integer	+	+	+	+	+	+	+	+	+	+	
enumerated	+	+	+	+	+	+	+	+	+	0 ₆	
real	+	0 ₄	-	+	+	+	0 ₄	+	0 ₄	0 ₆	
bitstring	+	+	0	+	+	+	+	+	+	0 ₆	
octetstring	+	+	+	+	+	+	+	+	+	0 ₆	
null	+	+	+	+	+	+	+	+	+	0 ₆	
sequence	+	+	+	+	+	+	+	0 ₂	+	0 ₆	
sequence-of	+	+	+	+	+	+	+	0 ₂	+	0 ₆	
set	+	+	+	+	+	+	+	0 ₂	+	0 ₆	
set-of	+	+	+	+	+	+	+	0 ₂	+	0 ₆	
choice	+	+	+	+	+	+	+	0 ₂	0 ₅	0 ₆	
selection	+	-	-	0 ₉	0 ₉	-	0 ₉	+	-	+	
tagged	+	+	+	+	+	0	+	+	0	+	
any	0 ₂	+	0 ₂	+	0 ₂	+	+	0 ₂	-	0 ₆	
object identifier	+	+	0 ₄	+	+	+	0 ₄	0 ₆	+	+	
character string	+	+	+	+	+	+	+	0 ₄	+	0 ₆	
single value	0 ₃	+	0 ₃	0 ₃	0 ₃	+	0 ₃	+	0 ₇	0 ₁	subtypes
contained	0 ₃	-	-	0 ₃	-	+	0 ₃	+	-	0 ₁	
value range	0 ₃	+	0 ₃	0 ₃	0 ₃	+	0 ₃	+	0 ₇	0 ₁	
size constraint	0 ₃	+	0 ₃	0 ₈	0 ₃	+	0 ₃	+	+	0 ₁	
perm. alphabet	0 ₃	+	-	0 ₃	0 ₃	+	0 ₃	+	+	0 ₁	
inner subtype	0 ₃	-	-	0 ₃	-	+	0 ₃	+	-	0 ₁	macro
macros	-	-	-	-	+	+	0	-	-	0 ₁	
ROS macros	-	0 ₅	-	-	+	+	0 ₅	-	+	+	
module def.	+	+	0	0 ₁₀	+	+	+	+	+	+	misc
import & export	+	+	0	-	+	+	0	+	+	0	
recursive types	+	+	+	+	+	+	-	+	-	+	
choice, any val.	{	{	S	:	S : {	S :	S	N	S	{	

- NOTE 1: This construct is ignored.
- NOTE 2: Value notation for this type not supported.
- NOTE 3: It is not checked whether values for this subtype satisfy the specified constraints.
- NOTE 4: Not all valid values are accepted for this type.
- NOTE 5: Not all valid instances of these macros are accepted.
- NOTE 6: Values for this type are ignored.
- NOTE 7: Only supported for the integer type.
- NOTE 8: Size constraint for sequence-of and set-of types are not supported.
- NOTE 9: The selection type can only be used as the "Type" within a "NamedType", not as a "NamedType" (see example 1).
- NOTE 10: A module cannot have an "AssignedIdentifier".
- NOTE 11: "Externaltypereference" and "Externalvaluereference" are not supported.
- NOTE 12: Instances without elements of this structured type are not accepted.

History

Document history	
October 1994	First Edition
March 1996	Converted into Adobe Acrobat Portable Document Format (PDF)