INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Z.100

Amendment 1
(10/2003)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

Formal description techniques (FDT) – Specification and Description Language (SDL)

Specification and Description Language (SDL)

Amendment 1: Backwards compatibility and compliance

ITU-T Recommendation Z.100 (2002) – Amendment 1

ITU-T Z-SERIES RECOMMENDATIONS

LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

FORMAL DESCRIPTION TECHNIQUES (FDT)	
Specification and Description Language (SDL)	**Z.100–Z.109**
Application of formal description techniques	Z.110–Z.119
Message Sequence Chart (MSC)	Z.120–Z.129
Extended Object Definition Language (eODL)	Z.130–Z.139
Testing and Test Control Notation (TTCN)	Z.140–Z.149
User Requirements Notation (URN)	Z.150–Z.159
PROGRAMMING LANGUAGES	
CHILL: The ITU-T high level language	Z.200–Z.209
MAN-MACHINE LANGUAGE	
General principles	Z.300–Z.309
Basic syntax and dialogue procedures	Z.310–Z.319
Extended MML for visual display terminals	Z.320–Z.329
Specification of the man-machine interface	Z.330–Z.349
Data-oriented human-machine interfaces	Z.350–Z.359
Human-computer interfaces for the management of telecommunications networks	Z.360–Z.369
QUALITY	
Quality of telecommunication software	Z.400–Z.409
Quality aspects of protocol-related Recommendations	Z.450–Z.459
METHODS	
Methods for validation and testing	Z.500–Z.519
MIDDLEWARE	
Distributed processing environment	Z.600–Z.609

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation Z.100

Specification and Description Language (SDL)

Amendment 1

Backwards compatibility and compliance

Summary

This amendment introduces new Annexes B and C. It also describes a number of minor changes to ITU-T Rec. Z.100 to link it to the new annexes and to correct and clarify some minor issues in the main body of ITU-T Rec. Z.100.

Annex B allows many legacy descriptions in SDL-92 to be recognized as valid according to SDL-2000.

Extensions to the grammar in the main body text of ITU-T Rec. Z.100 are defined.

Annex C defines what it means if a claim is made that a description is compliant with SDL or if a claim is made that a software tool is an SDL tool.

While in principle it is just sufficient to require descriptions to conform to SDL to enable them to be well-defined and reusable, in practice the notation actually used in models normally is determined by the tool used. For this reason the conformance of a tool to the language standard tends to be more useful than a statement that a model conforms to a standard. The intention is that a user can be confident that a description complies with the standard if it has been passed as compliant by a tool that satisfies the conformance rules.

There are specific additional requirements for Level 1 or Level 2 CIF support.

Source

Amendment 1 to ITU-T Recommendation Z.100 (2002) was approved on 29 October 2003 by ITU-T Study Group 17 (2001-2004) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation Z.100

Specification and Description Language (SDL)

Amendment 1

Backwards compatibility and compliance

1) Changes to underlying styles to support the Z.100 Microsoft Word macros

During the creation of the final published Z.100 document from the text approved by Study Group 17 the Microsoft style formats "z100 syntax example", "z100 syntax 1st line" and "z100 syntax last line" were renamed as "Topic Syntax example", "Topic Syntax cont." and "Topic Syntax" respectively. To support the Microsoft Word macros included in Z.100 and used for maintenance of the Z.100 document, the style names are changed back to the original names.

2) Update of Summary

In the first paragraph of sub-heading ***Status/Stability****, the following text is deleted:* ", but further study is required to complete Annex F".

In the first list under the sub-heading ***Status/Stability****, replace* "Reserved for future use – Annex B (03/93) is no longer valid" *by* "Backwards compatibility (added 2003)".

In the first list under the sub-heading ***Status/Stability****, replace* "Reserved for future use – Annex C (03/93) is no longer in force" *by* "Compliance to this Recommendation (added 2003)".

3) Update of clause 1.5 – Differences between SDL-92 and SDL-2000

In the list of keywords of SDL-92 that are not keywords in SDL-2000, the following keywords are deleted: **endnewtype**, **fpar**, **imported**, **newtype**, **returns**.

After this list, the following text is added:

"The following keywords of SDL-92 are keywords of SDL-2000:

endnewtype, **fpar**, **imported**, **newtype**, **returns**."

4) Correction in clause 6.1 – Lexical rules

<note text> *is corrected to* <comment text> *in*:

<comment body> ::=

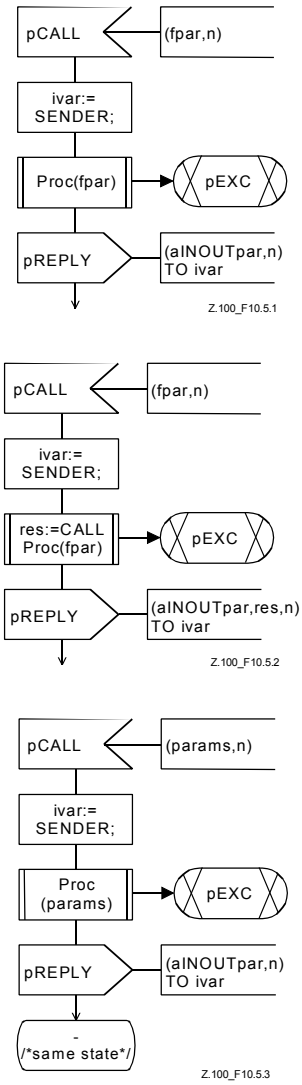
<solidus> <number sign> <comment text> <number sign>+ <solidus>

5) Corrections to diagrams in clause 10.5 – Remote procedures

Under Model sub-paragraph b) in the diagrams, "INOUT" *is replaced by "*aINOUT*" because "*INOUT*" is not mentioned in the text.*

Under Model sub-paragraph b) in the second diagram, "iva:=" *is corrected to* "ivar:=".

The corrected diagrams are:



6) New Annex B

Annex B

Backwards compatibility

SDL-2000 introduces some syntactic changes to SDL that invalidate descriptions written for older versions of SDL supported by tools that were available prior SDL-2000 being approved. In most cases the models defined by these legacy descriptions, if analysed using an appropriate concrete grammar, can be interpreted as SDL-2000 without any change to the behaviour of the model. The purpose of this annex is to define the concrete grammar for models defined using tools which supported older versions of SDL, including models produced by tools supporting a version of SDL based on SDL-92 but supporting some SDL-2000 features.

The grammar defined in this annex therefore extends the notation allowed for SDL-2000 without extending the semantics. This grammar allows SDL-2000 tools to provide backwards compatibility for older descriptions and allows older tools to be used as tools for a subset of the SDL-2000 language. Moreover, as the Recommendation for SDL-2000 replaces the previous SDL

Recommendation as the Recommendation in force, this annex is needed so that existing valid SDL descriptions using SDL-92 remain valid SDL descriptions.

B.1 Background

The rationale for some changes to the SDL language grammar was to provide a concrete grammar for the language that could be more easily understood by 21st-century engineers who have learnt programming languages such as C++ and Java. Some idiosyncratic keywords such as **fpar** and syntax such as headings were therefore changed. The SDL-92 syntax for such constructs can easily be mapped into SDL-2000.

The semantics supported by SDL-2000 differs from previous versions of SDL. There are some features of SDL-92 that are not supported in SDL-2000. Models that use view expression, generators, block substructure, channel substructure, signal refinement or axiomatic definition of data are not compatible with the semantics of SDL-2000, and the grammar for these features is not included in this annex. SDL-2000 has limited support for graphical macros, and models that used advanced features for graphical macros in SDL-92 are not supported. Some of these features, such as the advanced graphical macros or models with implicit channels and signal routes, were not well supported by tools, so the machine-readable SDL files using these features are unlikely to exist. For this reason, these features of models in SDL-92 have to be rewritten for use with SDL-2000.

On the other hand SDL-2000 has features that are not supported by SDL-92 or tools such as composite states. To use these features when modifying an existing model, the model should be converted to SDL-2000. Appendix III describes an approach to the systematic conversion of SDL-92 to SDL-2000.

B.2 Lexical rules

Previous versions of SDL were not case sensitive. Keywords of the language could be in mixed case, and different occurrences of the same name could have a different case mix. Although tools may support a mode where they are case insensitive, a model that is not case correct in the spelling of keywords and inconsistent in the case usage for a name is not valid according to this annex. The model is required to be case correct according to SDL-2000 lexical rules and national characters or spaces in names are not permitted.

This annex extends <keyword> to include:

all | endnewtype | fpar | newtype | returns

B.3 Macro

Concrete grammar

The syntax rule <agent signature> is extended to allow **fpar** syntax.

<macro formal parameters> ::=
(<macro formal parameter> { **,** <macro formal parameter>}* **)**
| **fpar** <macro formal parameter**>** {, <macro formal parameter>}*

B.4 Context parameters

The signatures of processes and procedures are different in SDL-92: the keyword **fpar** is used to introduce the list of sorts and a procedure result is introduced by the keyword **returns**.

B.4.1 Agent context parameter

Concrete grammar

The syntax rule <agent signature> is extended to allow **fpar** syntax.

<agent signature> ::=
<sort list>
| [<end>] **fpar** <sort> {, <sort> }

B.4.2 Procedure context parameter

Concrete grammar

The syntax rule <procedure constraint> is extended to allow the <legacy procedure signature>.

<procedure constraint> ::=
atleast <procedure identifier>
| <procedure signature in constraint>
| <legacy procedure signature>
<legacy procedure signature> ::=
[[<end>] **fpar** <formal parameter> {, <formal parameter> }*
[<end> **returns** <sort>]]
| [<end>] **returns** <sort>

B.4.3 Sort context parameter

Concrete grammar

The syntax rule <sort context parameter> is extended to allow the keyword **newtype**, which has the same meaning as **value type**.

<sort context parameter> ::=
{ { **value** | **object** } **type** | **newtype** } <sort name> <sort constraint>

NOTE – The legacy <sort signature> is not supported.

B.5 Agents

Concrete grammar

The syntax rule <agent formal parameters> is extended to allow the formal parameters to be specified with **fpar**.

<agent formal parameters> ::=
(<parameters of sort> {, <parameters of sort>}* **)**
| [<end>] **fpar** <parameters of sort> {, <parameters of sort>}*

NOTE – The optional <end> before the keyword **fpar** is added to validate models to be defined using tools that require a semicolon at this point, even though it was only valid in SDL/GR in SDL-92 if a <number of instances> was included.

SDL-92 required an <imported procedure specification> in each process or process type that used a remote procedure, whereas in SDL-2000 it is sufficient for the remote procedure definition to be visible in the process or process type. SDL-92 also required an <imported variable specification> in each process or process type that used a remote variable, whereas in SDL-2000 it is sufficient for the remote variable definition to be visible in the process or process type. The syntax is extended to allow <imported procedure specification> and <imported variable specification>: the syntax rule <agent text area> is extended to become:

<agent text area> ::=
<text symbol>
contains {
[<valid input signal set>]
{ <signal definition>
| <signal reference>
| <signal list definition>
| <variable definition>
| <remote procedure definition>
| <remote variable definition>
| <data definition>
| <data type reference>
| <timer definition>
| <imported procedure specification>
| <imported variable specification>
| <interface reference>
| <macro definition>
| <exception definition>
| <procedure definition>
| <procedure reference>
| <select definition>
| <agent type reference>
| <agent reference> }* }

<imported procedure specification> ::=
imported procedure <remote procedure identifier> <end>
[<legacy procedure signature> <end>]

Model

An <imported procedure specification> has no SDL meaning and is treated as a comment, though to be compatible with SDL-92 the <remote procedure identifier> should refer to a remote procedure that is consistent with the <formal parameter>s and returned <sort>.

B.6 Procedure

Concrete grammar

The syntax for <procedure formal parameters> is extended to allow the formal parameters to be specified with **fpar**.

<procedure formal parameters> ::=
(<formal variable parameters> {, <formal variable parameters> }*)
| [<end>] **fpar** <formal variable parameters> {, <formal variable parameters> }*

NOTE – The optional <end> before the keyword **fpar** is added to validate models to be defined using tools that required a semicolon at this point, even though it was not valid in SDL/GR in SDL-92.

The syntax for <procedure result> is extended to allow specification with **returns**.

<procedure result> ::=
<result sign> [<variable name>] <sort>
| **returns** [<variable name>] <sort>

B.7 Remote variables

Concrete grammar

The rule <remote variable definition> is extended to allow the keyword **nodelay**.

<remote variable definition> ::=
remote <remote variable name> {**,**<remote variable name>}* <sort> [**nodelay**]
{**,** <remote variable name> {**,** <remote variable name>}* <sort> [**nodelay**]}* <end>
<imported variable specification> ::=
imported
<remote variable identifier> {, <remote variable identifier> }* <sort>
{, <remote variable identifier> {, <remote variable identifier> }* <sort>}* <end>

Model

The keyword **nodelay** has no SDL-2000 meaning, though to be compatible with SDL-92 the channel conveying the signals for the remote variable should be a channel without delay.

An <imported variable specification> has no SDL-2000 meaning and is treated as comment, though to be compatible with SDL-92 the <remote variable identifier> should refer to a remote procedure that is consistent with the <formal parameter>s and returned <sort>.

B.8 Specialization of data types

The operators inherited in SDL-92 are specified, whereas in SDL-2000 all the visible operators are inherited.

Concrete grammar

The syntax for <data type specialization> is extended to allow specification according to SDL-92 syntax as follows:

<data type specialization> ::=
inherits <data type type expression> [<renaming> | <legacy data inheritance>] [**adding**]
<legacy data inheritance> ::=
[<legacy literal renaming>]
[[**operators**] { **all** | (<legacy inheritance list>) } [<end>]]
<legacy literal renaming> ::=
literals <rename pair> { , <rename pair> }* <end>

To be consistent with SDL-92, the first name in the <rename pair> in a <legacy literal renaming> should refer to a literal defined in the base type.

<legacy inheritance list> ::=
<legacy inherited operator> { , <legacy inherited operator> }*
<legacy inherited operator> ::=
<operation name> | <rename pair>

To be consistent with SDL-92, the <operation name> or the first name in the <rename pair> in a <legacy literal renaming> should refer to an operation defined in the base type. Specifying **operators all** or a named operation without renaming has no influence on the inherited operations, which are determined according to the SDL-2000 rules.

B.9 Behaviour of operations

Concrete grammar

The syntax for <operation result> is extended to allow specification with **returns**.

<operation result> ::=
<result sign> [<variable name>] <sort>
| **returns** [<variable name>] <sort>

B.10 Optional definition

Concrete grammar

To be compatible with SDL-92 models, the syntax is extended to allow <imported variable specification> and <imported procedure specification> in a <select definition>.

```
<select definition> ::=
                    select if ( <Boolean simple expression> ) <end>
                    {           <agent type reference>
                          |     <agent reference>
                          |     <signal definition>
                          |     <signal list definition>
                          |     <signal reference>
                          |     <remote variable definition>
                          |     <remote procedure definition>
                          |     <data definition>
                          |     <data type reference>
                          |     <interface reference>
                          |     <timer definition>
                          |     <variable definition>
                          |     <imported variable specification>
                          |     <procedure definition>
                          |     <imported procedure specification>
                          |     <procedure reference>
                          |     <select definition>
                          |     <macro definition>
                          |     <exception definition> }+
                    endselect <end>
```

7) New Annex C

Annex C

Compliance to this Recommendation

Scope

Descriptions that claim to be compliant to this Recommendation (SDL) shall conform to the notation grammar defined by this Recommendation and ITU-T Recs Z.105 and Z.106 with the semantics as defined in these Recommendations. A description is non-compliant if it includes notation grammar that is not allowed by these Recommendations, or has analysable semantics that can be shown to differ from these Recommendations.

A software tool that claims to support Z.100 (in the following called a tool) should be capable of creating, editing, presenting and analysing descriptions compliant with this Recommendation.

C.1 Definitions of valid tools

C.1.1 compliant SDL tool: A tool that detects non-compliance of a description with ITU-T Rec. Z.100. If the tool handles a superset notation, it is allowed to categorize non-compliance as a warning rather than a failure.

C.1.2 fully compliant SDL tool: A compliant SDL tool that supports the complete grammar defined by ITU-T Recs Z.100, Z.105 and Z.106.

C.1.3 valid Z.100 SDL tool: A compliant SDL tool that supports the graphical SDL grammar defined in ITU-T Rec. Z.100.

C.1.4 valid Z.100 SDL with ASN.1 tool: A valid Z.100 SDL tool that also supports ASN.1 as modules according to ITU-T Rec. Z.105.

C.1.5 valid Z.106 SDL tool: A compliant SDL tool that supports the textual SDL grammar as defined in Level 0 CIF (clause 5/Z.106), which, by definition, includes the semantics and some concrete syntax of ITU-T Rec. Z.100.

C.1.6 valid Z.106 SDL with ASN.1 tool: A valid Z.106 SDL tool that also supports ASN.1 as modules according to ITU-T Rec. Z.105.

C.2 Conformance

A conformance statement clearly identifying the language features and requirements not supported should accompany any tool that handles a subset of this Recommendation and ITU-T Recs Z.105 and Z.106. If no conformance statement is provided, it shall be assumed that the tool is a fully compliant SDL tool. It is therefore preferable to supply a conformance statement; otherwise, any unsupported feature allows the tool to be rejected as not valid.

8) Replacement text for Appendix I – Status of Z.100, related documents and Recommendations

Replace the text of Appendix I with the following:

This appendix contains a list of the status of SDL-related Recommendations issued by ITU-T. The list includes all parts of this Recommendation and of ITU-T Recs Z.105, Z.106, Z.107, Z.109 and any related methodology documents. It also lists other relevant Recommendations such as ITU-T Rec. Z.110.

This list shall be updated by appropriate means (for example, a corrigendum) whenever changes to SDL are agreed and new Recommendations approved.

SDL-2000 is defined by the following Recommendations approved by ITU-T Study Group 17 on August 2002 except as documented below.

– ITU-T Recommendation Z.100, *Specification and Description Language (SDL)*.

– Annex A to ITU-T Recommendation Z.100, *Index of non-terminals*.

– Annex B to ITU-T Recommendation Z.100 (2003), *Backwards compatibility*.

– Annex C to ITU-T Recommendation Z.100 (2003), *Compliance.*

– Annex D to ITU-T Recommendation Z.100, *SDL predefined data*.

– Annex F to ITU-T Recommendation Z.100 (2000), *SDL formal definition* (approved by ITU-T Study Group 10 on 24 November 2000).

  Tools for the formal semantics reference model of SDL-2000 (ITU-T Specification and Description Language) can be found at http://sourceforge.net/projects/sdlc (the files themselves are accessible either through CVS, or through the CVS web front end, at http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/sdlc/SDLC/).

No specific plans at the time of approval for Annex E.

– Supplement 1 to ITU-T Recommendation Z.100 (1997), *SDL+ methodology: use of MSC and SDL (with ASN.1)*.

– ITU-T Recommendation Z.105 (2003), *SDL combined with ASN.1 modules (SDL/ASN.1)*.

– ITU-T Recommendation Z.106 (2002), *Common interchange format for SDL*.

– ITU-T Recommendation Z.107 (1999), *SDL with embedded ASN.1*.

– ITU-T Recommendation Z.109 (1999), *SDL combined with UML*.

– ITU-T Recommendation Z.110 (2000), *Criteria for the use of formal description techniques by ITU-T* (approved by ITU-T Study Group 10 on 24 November 2000).

Further information on SDL including information on books and other publications is available via: http://www.sdl-forum.org/.

9) Clarifications in Appendix II – Guidelines for the maintenance of SDL

In II.1.3, Change request procedure, the term "clarification" *is changed to* "clarification (or question)" *throughout including the figure.*

The text "Study Group 10 experts" *is changed to* "experts in the Study Group responsible for SDL".

10) Corrections in Appendix III – Systematic conversion of SDL-92 to SDL-2000

In item 3, "fpar" *is corrected to appear in bold face and* "return" *is replaced by* "**returns**".

In item 5, "add implicit channels" *is corrected to* "add explicit channels".

In item 5, "Addendum 1" *is replaced (as a clarification) by* "Addendum 1 to SDL-92 (1996)".

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure, Internet protocol aspects and Next Generation Networks
Series Z	**Languages and general software aspects for telecommunication systems**

Geneva, 2004