Chapter

55 *The SDL Analyzer*

The SDL Analyzer's primary task is to check SDL-92 diagrams (extended as defined in "Compatibility with ITU SDL" on page 9 in chapter 1, *Compatibility Notes, in the Release Guide*) with respect to syntactic and semantic rules. The Analyzer accepts both SDL/GR and SDL/PR as input, and has the ability to transform SDL/GR diagrams to SDL/PR files and vice-versa.

This chapter is a reference to the Analyzer commands. Its limitations according to the Z.100 recommendation is found in "SDL Analyzer" on page 37 in chapter 2, *Release Notes, in the Release Guide*.

For a guide to how to perform syntactic and semantic check on an SDL diagram structure, see chapter 56, *Analyzing a System*. This chapter also gives an overview of the Analyzer.

The Analyzer User Interfaces

The Analyzer Graphical UI

The Analyzer's graphical user interface is managed by the Organizer.

To start the Analyzer, select *Analyze* from the Organizer *Generate* menu or click the *Analyze* quick button. For more information, see "Analyze" on page 111 in chapter 2, *The Organizer* and "Quick Buttons" on page 176 in chapter 2, *The Organizer*.

The Analyzer's graphical user interface is further described in "Using the Analyzer" on page 2548 in chapter 56, *Analyzing a System*.

The Analyzer Batch UI

See *"Batch Facilities" on page 203 in chapter 2, The Organizer* for information on running the Analyzer non-interactively.

The Analyzer Command-Line UI

The Analyzer also has a *command-line mode*, when it is started from the OS outside the graphical environment of the SDL suite. This is the user interface that is described in the remainder of this chapter.

Starting the Analyzer

The Analyzer is started in command-line mode from the OS prompt with the command:

```
sdtsan [1]
```

When the Analyzer is started in command-line mode, it responds with the prompt

```
SDL Analyzer
Command :
```

1. The directory where the SDL suite binaries are stored is assumed to be included in the PATH variable. Otherwise, the complete directory path must be supplied.

Environment Variables

- SDTSAN_WARN_GATE

  By setting this to any value the analyzer will give warnings instead of errors for unconnected gates that should be connected. This option is only provided for backwards compatibility and should not be used if possible.

- SDTSAN_ASN1_ENUM_OP

  By setting this to any value the analyzer will pass the -e option to asn1util. This will create all operators for ASN.1 ENUMERATED types as defined in Z.105.

- SDTSAN_FILTER_USE

  By setting this to any value the analyzer will run the specified filter command specified in the analyzer options once before SDL package dependencies are resolved. This makes it possible for external filter scripts to exclude SDL package before make files are generated.

Syntax of Analyzer Commands

A command name may be abbreviated by giving sufficient characters to distinguish it from other command names. A hyphen ('-') is used to separate command names into distinct parts.

Any part may be abbreviated as long as the command does not become ambiguous.

Parameters to commands are separated by one or several spaces. In case you omit a parameter of type "on/off", the value currently defined will toggle between its "on" and "off" values.

In both command names and parameters there is no distinction made between upper and lower case letters.

> **Note:**
>
> **On UNIX** systems, distinction is made between upper case letters and lower case letters for file names.

Comments are indicated by the "#" character, and extend to the end of the line. Comments may only be preceded by white space.

Description of Analyzer Commands

In this section, the Analyzer commands are listed in alphabetical order, along with their parameters and a description.

! (shell escape)

Parameters:

```
<operating system command>
```

Escape to the shell to do command.

Add-Input

Parameters:

```
<Filespec>
```

Add a file to the list of files to be converted to PR. The file will be processed according to the setting of Input-Mode.

Analyze

Parameters:

```
(None)
```

This command orders the Analyzer to perform the passes according to the *Analyzer options* as they are defined.

> **Hint:**
>
> The current analysis options can be printed using the command *Show-Analyze-Options* (see "Show-Analyze-Options" on page 2426).

The order in which the passes are performed is:

1. *Conversion to PR*, if the input is C header, ASN.1 or SDL/GR diagrams.
2. *Macro expansion* (optional).
3. *Syntactic analysis* (optional).
4. *Pretty-printing* of a PR file (optional).
5. *PR to GR conversion* (optional). This pass converts an SDL/PR input file to SDL/GR diagrams, using the native format of the SDL suite.

6. *Semantic analysis* (optional).
7. *Cross reference generation* (optional).

Note:

The Analyzer may need to be reset (using the *New* or *Clear* command) between subsequent *Analyze* commands, since the Analyzer keeps track of what passes have been performed in order to minimize analysis time.

ASN1-Coder-Name

Parameters:

```
<name>
```

The prefix of encode and decode functions for ASN.1 types. Default is BER.

ASN1-Keyword-File

Parameters:

```
<file>
```

Change keywords in files output by asn1util according to file. Default is no file, in this case the keywords in file asn1util_kwd.txt in the SDT installation is used.

Asn1Util

Parameters:

```
<options>
```

Invoke asn1util through the post master. Same options as the command line interface.

Cd

Parameters:

```
<directory>
```

Change current working directory to <directory>.

Clear

Parameters:

```
(None)
```

Force the Analyzer to perform all passes from scratch as defined in the Analyzer options on the previously selected input. See also the *New* command described in "New" on page 2414.

Coder-Buffer-In-Sdl

Parameters:

```
Octet-String | Coder-Buf | Custom-Coder-Buf | None
```

Pick representation of ASN.1 Buffer in SDL, None means no encoding available in SDL. Default is Octet-String. For more information see "Buffer Management System" on page 2776 in chapter 59, *ASN.1 Encoding and De-coding in the SDL Suite*.

ComplexityMeasurement-File

Parameters:

```
[<file spec>]
```

Specifies where the complexity measurements file is stored if the Set-ComplexityMeasurement command is set.

Default is `<system name>.csv`.

See chapter 49, *Complexity Measurements* for more information.

Component

Parameters:

```
[<SDL qualifier>]
```

Add a new item to a program. See *"Partitioning" on page 2571 in chapter 57, The Cadvanced/Cbasic SDL to C Compiler*.

Cpp2sdl

Parameters:

<options>

Invoke cpp2sdl through the post master. Same options as the command line interface.

Env-Header-Channel-Name

Parameters:

```
[<name>]
```

Configuration of channel names in environment header file (.ifc). %n is the name of the entity in SDL, %s is the name of the scope. Omit <name> to exclude the definitions completely from the file. Default is %n.

Env-Header-Literal-Name

Parameters:

`[<name>]`

Configuration of literal names in environment header file (.ifc). %n is the name of the entity in SDL, %s is the name of the scope. Omit <name> to exclude the definitions completely from the file. Default is %n.

Env-Header-Operators

Parameters:

`[On/Off]`

Configure if operators should be present in environment header file (.ifc). Default is On.

Env-Header-Signal-Name

Parameters:

`[<name>]`

Configuration of signal names in environment header file (.ifc). %n is the name of the entity in SDL, %s is the name of the scope. Omit <name> to exclude the definitions completely from the file. Default is %n.

Env-Header-Synonym-Name

Parameters:

`[<name>]`

Configuration of synonym names in environment header file (.ifc). %n is the name of the entity in SDL, %s is the name of the scope. Omit <name> to exclude the definitions completely from the file. Default is %n.

Env-Header-Type-Name

Parameters:

`[<name>]`

Configuration of type names in environment header file (.ifc). %n is the name of the entity in SDL, %s is the name of the scope. Type definitions are always present in the file. Default is %n.

Error-File

Parameters:

```
[<file spec>]
```

An optional file to store analyzer messages in. Default is no error file.

Exit

Parameters:

```
(None)
```

Terminates the Analyzer and returns you to the Operating System prompt.

Filter

Parameters:

```
[Command]
```

This command allows filtering or preprocessing files before they are read by the Analyzer. The command should be an executable file, preferably found in $PATH. The Analyzer appends three parameters to the supplied command. The first parameter is the name of the file, the second parameter tells what the Analyzer is about to do with the file, and the third is the Source Directory. Currently the following are available:

- **use** (before SDL package dependencies are resolved, this phase is only activated when the environment variable SDTSAN_FILTER_USE is set)
- **import** (before converting to SDL PR)
- **macro** (before macro expansion)
- **parse** (before syntax analysis)
- **include** (before syntax analysis)

The command is supposed to modify the file given in the parameter and leave the result in the same file. Note that the files passed are your original files.

Generate-Advanced-C

Parameters:

```
(None)
```

This command invokes the Cadvanced SDL to C Compiler, which will produce one / multiple C program(s), as well as a makefile. This C program is then compiled and linked in order to build up an executable ap-

plication. The Cadvanced SDL to C Compiler uses the options which have been previously defined using the various Generate Options commands. See also Generate-Basic-C.

Generate-Basic-C

Parameters:

`(None)`

This command invokes the Cbasic SDL to C Compiler, which will produce one / multiple C program(s), as well as a makefile. This C program is then compiled and linked in order to build up an executable application. The Cbasic SDL to C Compiler uses the options which have been previously defined using the various Generate Options commands. This command cannot be used when building applications, but otherwise it is identical to Generate-Advanced-C.

Generate-Micro-C

Parameters:[1]

`(None)`

This command invokes the Cmicro SDL to C Compiler, which will produce one / multiple C program(s), as well as a makefile. This C program is then compiled and linked in order to build up an executable application. The Cmicro SDL to C Compiler uses the options which have been previously defined using the various Generate Options commands. See also Generate-Advanced-C.

GR-PR-File

Parameters:

`[<file spec>]`

This command specifies where the output of the conversion to PR pass is stored. Default is `<infile>.pr`.

GR2PR

Parameters:

`<GR-File> <PR-File>`

The command converts the file designated by the file specifier `GR-File` to a textual (i.e. PR) description of the diagram, storing the results in the file designated by the file specifier `PR-File`.

1. The exact syntax is subject to change since the tool is still under development.

The file GR-File should contain an SDL diagram stored in binary format.

Help

Parameters:

[Command]

The command issues on-line help for the Analyzer commands. Depending on if a command is specified or not, the following happens:

- By typing this command without parameters, the available commands are listed with their parameters. Control is then transferred to the Analyzer on-line help, where additional information about all commands is available. The on-line help prompt looks like this:

  Topic?

  – Typing a command name followed by <Enter> issues information about the command.

  – Typing <Enter> only, returns control to the Analyzer prompt.

- Typing the Help command followed by a command name, on-line for that command will be displayed. Control is then transferred to the Analyzer on-line help utility.

You can abbreviate any topic name, although ambiguous abbreviations result in all matches being displayed.

Include-Directory

Parameters:

[<directory>]

Where the analyzer looks for include files. An empty <directory> clears the list.Subsequent commands adds more directories to the list. Default is an empty list.

Include-File

Parameters:

<file spec>

Execute analyzer commands in <file spec>.

Include-Map

Parameters:

```
<logical name> <file spec>
```

Maps logical names to file names for graphical includes when converting graphical SDL (GR) to text (PR). This command should be preceded by commands to specify input mode to GR and a GR file. Subsequent commands adds more maps to the list. Default is an empty list.

Input-Mode

Parameters:

```
ASN.1 | C-Header | GR | PR
```

Select filter to convert input to PR. Default is PR.

Instance-File

Parameters:

```
[<file spec>]
```

Where the instance tree is stored. Default is `<system name>.ins`. See "SDL Instance Information" on page 2442 for more information.

Macro-PR-File

Parameters:

```
[<file spec>]
```

This command specifies where the output of the macro expander is stored. Default is `<infile>.prm`.

Make-File

Parameters:

```
[<file spec>]
```

Name of makefile. Default is `<system name>.m`.

Make-Template-File

Parameters:

```
[<file spec>]
```

Name of make template file. Default is no template.

New

Parameters:

```
(None)
```

This command initializes the Analyzer. Following the *New* command, you will need to specify an input to the Analyzer using the *Set-Input* command.

See also "Clear" on page 2407.

Operating-System

Parameters:

```
(None)
```

Creates a subprocess which returns you temporarily to the Operating System. Once you terminate this process, control is returned to the Analyzer.

Organizer-Object

Parameters:

```
Level Type Name File Sep SepName Selected [Depend]
```

Pass the Organizers info on a diagram to the Analyzer. This command is used by the Organizer.

Pretty-PR-File

Parameters:

```
<Filespec>
```

This command specifies where the output of the pretty printer is stored. Default is `<infile>.sdl`.

Program

Parameters:

```
[<name>]
```

Name of program built from components. See *"Partitioning" on page 2571 in chapter 57, The Cadvanced/Cbasic SDL to C Compiler*.

Quit

Parameters:

```
(None)
```

Synonym for exit.

SDL-Coder-Name

Parameters:

```
<name>
```

The prefix of encode and decode functions for SDL types. Default is ASCII.

SDL-Keyword-File

Parameters:

```
[<file spec>]
```

Produce a list of SDL keywords and SDT Ref in a file. Default is no list.

SDT-Ref

Parameters:

```
[<sdt ref>]
```

Use <sdt ref> in error messages from command parser overriding any other <sdt ref>.

SDT-SYSTEM-4.5

Parameters:

```
(None)
```

The Organizer uses this to make sure the right version of the analyzer is running.

Set-ASN1-Coder

Parameters:

```
[On/Off]
```

Generates encoder and decoder from ASN.1 modules (requires separate license).

Default is off.

Set-C-Compiler-Driver

Parameters:

```
[On/Off]
```

This option informs the SDL to C compilers if the SDL C Compiler Driver should be invoked. See chapter 61, *SDL C Compiler Driver (SC-CD)*.

Default is off.

Set-C-Plus-Plus

Parameters:

`[On/Off]`

SDL to C compilers generate code to be compiled with C++ compiler if on. Default is off.

Set-Case-Sensitive

Parameters:

`[On/Off]`

Case sensitive SDL names if On. Keywords must be all upper or lower case if on.

Default is off.

Set-Compile-Link

Parameters:

`[On/Off]`

This option informs the Code Generator (SDL to C compilers) whether the generated code should be compiled and linked automatically (by executing the generated makefile).

Default is on.

Set-ComplexityMeasurement

Parameters:

`[On/Off]`

This option will generate a complexity measurement file. Following the *Set-ComplexityMeasurement* command, you may specify a file name using the ComplexityMeasurement-File command.

Default is off.

See chapter 49, *Complexity Measurements* for more information.

Set-Echo

Parameters:

`[On/Off]`

This option will print (echo) all Analyzer commands as they are executed. Default is off.

Set-Env-Function

Parameters:

`[On/Off]`

This option informs the code generators (SDL to C compilers) if environment functions should be generated or not.

Default is off.

Set-Env-Header

Parameters:

`[On/Off]`

This option is applicable to the SDL to C compilers only. With this parameter turned on, a header file containing the definitions of the SDL system's interface to the environment is created. This file is identified by the `.ifc` file name extension.

Default is off.

Set-Error-Limit

Parameters:

`<Errorlimit>`

This command sets the limit of reported errors after which the Analyzer will stop its execution. Use 0 to turn off the limit.

Default is a limit of 30 diagnostics.

Set-Expand-Include

Parameters:

`[On/Off]`

This option specifies whether include directives should be handled and expanded when reading input files. See "Including PR Files" on page 2436.

Default is on.

Set-Expression-Limit

Parameters:

`<Depth>`

This command instructs the Analyzer to issue a warning when the semantic analyzer encounters an SDL expression which is at least *Depth* deep.

> **Note:**
>
> Deeply nested expressions may cause a **significant degradation of performance** when performing the semantic analysis pass. It is therefore recommended to break down complex expressions into multiple, less complex expressions.

Default is 0, meaning that no warnings will be issued.

Set-File-Prefix

Parameters:

```
[<file name prefix>]
```

Prefix for files generated by the code generators. Default is no prefix.

Set-Full

Parameters:

```
[On/Off]
```

Redo everything if on. Default is off.

Set-Generate-All-Files

Parameters:

```
[On/Off]
```

Existing files will not be touched by C code generators if the new version is identical to the existing one and this option if Off. Default is On.

Set-Ignore-Hidden

Parameters:

```
[On/Off]
```

Decides if PR should be generated for hidden SDL symbols. Default is On.

Set-Implicit-Type-Conversion

Parameters:

```
[On/Off]
```

When this option is on, the Analyzer allows implicit type conversion of reference data types (generators Own, ORef, and Ref). Note that analyzing large expressions with this option on is slow. Default is off.

For more information, see "Implicit Type Conversions" on page 134 in chapter 3, *Using SDL Extensions, in the SDL Suite Methodology Guidelines*.

Set-Input

Parameters:

`<Filespec>`

The Analyzer will use the file specified in filespec as input file. The file will be processed according to the setting of Input-Mode (see "Input-Mode" on page 2413). Use Add-Input to process a system stored in several files.

Set-Instance

Parameters:

`[On/Off]`

With this option on, a file containing the SDL instance tree will be generated. Default is off. See "SDL Instance Information" on page 2442 for more information.

Set-Kernel

Parameters:

`<Kernel>`

This option is applicable to the SDL to C compilers only. This option allows you to select the appropriate kernel with predefined runtime libraries. The makefile generated by the SDL to C compilers contains instructions which link the generated and compiled code with object modules of the selected library. The effect of this will be that the code which is generated by the SDL to C compilers will have different properties, reflecting the purpose of its usage (simulation, application generation, etc.). The allowed values of the parameter kernel vary from one configuration to another. The definitions are made in the file `sdtsct.knl`. The definitions for Cadvanced are made in the file `sdtsct.knl`, whereas the definitions for Cmicro are made in the file `scmsct.knl`.

The following values are recognized for Cadvanced:

Library	Application area
`SCTADEBCOM`	Simulation, simulated time
`SCTADEBCLCOM`	Simulation, real time
`SCTAPERFSIM`	Performance simulation
`SCTAAPPLCLENV`	Application

Library	Application area
SCTADEBCLENV	Application debug (simulation with environment, only Cadvanced)
SCTAVALIDATOR	Validation
SCTATTCNLINK	TTCN link

The following values are recognized for Cmicro:

Library	Application area
SCMADEBCOM	SDL Target Tester and communication, no real time clock
SCMADEBCLCOM	SDL Target Tester, communication and real time clock
SCMADEBCLENVCOM	SDL Target Tester, communication, user's env and real time clock
SCMAAPPLCLENVMIN	Application with environment, clock, etc.
SCMAAPPLCLENV	Evaluation Kernel (no complex ADT; no SDL Target Tester)

Note:

The predefined kernels for Cmicro cannot be used with any C compiler. They require GCC 2.8.1 **on UNIX**, or Borland C 5.02 or Microsoft Visual C++ 5.0 **in Windows**.

Set-Lower-Case

Parameters:

`[On/Off]`

SDL does not distinguish between lower case letters (a..z) and upper case letters (A..Z), but the high level programming language C does. With this parameter turned on, the names of, for instance, variables in the generated C code is written in lower case letters only. Otherwise, the case of an item in the SDL definition is used.

Default is off.

Set-Macro

Parameters:

`[On/Off]`

Turns on or off the macro expansion pass. This option must be turned on if your diagram contains references to macro(s).

Default is off.

Set-Make

Parameters:

`[On/Off]`

This command affects an option which enables or disables the generation of a makefile in conjunction with generation of C code.

Default is on.

Set-Modularity

Parameters:

`[Modularity]`

Allows to specify the modularity of the code generated by the code generators, e.g. the SDL to C compilers. This option affects how many files will be generated.

Three options for the parameter *Modularity* are available:

- *No* – One single code module is generated.
- *User* – User defined, according to the definitions in the Organizer. See "Edit Separation" on page 136 in chapter 2, *The Organizer*.
- *Full* – Multiple code modules will be generated.

Default value is *No*.

Set-Optional-Make-Operator

Parameters:

`[On/Off]`

Optional and default struct parameters are included in the make operator (invoked by (. .)) if On. Default is Off.

Set-Output

Parameters:

`<Filespec>`

Allows you to save the information generated by the Analyzer in files with a different name, using *Filespec* as the base name.

File name extensions are appended automatically by the Analyzer.

Set-Pr2Gr

Parameters:

`[On/Off]`

Turning this option on, the Analyzer will perform the PR to GR conversion pass. The steps are:

1. The Analyzer creates diagram files with temporary names.
2. The Analyzer lets the SDL Editor do tidy up on the diagram files.
3. The Organizer renames/moves diagram files according to its preference.

Step 3 is only performed when a PR to GR conversion is run from the Organizer. The request to put temporary files in the current working directory is ignored on the Windows platform (they end up in some temp directory).

Default is off.

Set-Predefined-XRef

Parameters:

`[On/Off]`

Include predefined data in SDL cross reference file if On. Default is Off.

Set-Prefix

Parameters:

`[Prefix]`

Allows you to specify what prefix the code generators add to SDL names in the generated code.

Possible values for the parameter *prefix* are:

- *No*
- *Entity*

- *Full*
- *Special*

Default is *Full*.

Set-Pretty

Parameters:

`[On/Off]`

Turning this switch on, a pretty printed PR file will be generated.

Default is off.

Set-References

Parameters:

`[On/Off]`

This option allows you to disable the verification that exactly one reference corresponds to each remote definition.

Default is on.

Set-SDL-Coder

Parameters:

`[On/Off]`

Generate encoder and decoder from SDL (requires separate license).

Default is off.

Set-Sdt-Ref

Parameters:

`[On/Off]`

Decides if SDT references should be included in the generated code as comments for easier navigation to its source. As default, SDT references in comments are included. Note that omitting SDT references in comments by using this command is not supported by standard Telelogic code generators.

Set-Semantic

Parameters:

`[On/Off]`

With this option turned on, the semantic check pass will be performed. Otherwise, the Analyzer will stop after the syntactic check.

Default is on.

See also "Optimizing a System to Reduce Analysis Time" on page 2551 in chapter 56, *Analyzing a System*.

Set-Signal-Number

Parameters:

`[On/Off]`

Code generators will generate a file with signal numbers if on.

Default is off

Set-Source

Parameters:

`[On/Off]`

This command informs the code generators whether to generate C code or not when a generate code command is given. The reason for disabling code generation is that you may want to produce a makefile only, without generating code.

Default is on.

Set-Synonym

Parameters:

`[On/Off]`

Decides if the specified synonym file should be used or not. Default is Off.

Set-Syntax

Parameters:

`[On/Off]`

This option turns on or off the syntax analysis pass.

Default is on.

Set-TAEX-Make

Parameters:

`[On/Off]`

This command controls the generation of a makefile for Targeting Expert in conjunction to generation of C code.

Default is off.

Set-Upcase-Keyword-Pretty

Parameters:

`[On/Off]`

Select between upper or lower case keywords in pretty printings.

Default is off (i.e. lower case letters).

Set-Warn-Else-Answer

Parameters:

`[On/Off]`

Print a message if a decision or transition option does not have an else answer and the Analyzer is unable to decide that it is not needed.

Default is on.

Set-Warn-Match-Answer

Parameters:

`[On/Off]`

Print a message if a decision or transition option does not have an answer for some value. The analyzer is not able to do this in all situations.

Default is on.

Set-Warn-Optional-Parameter

Parameters:

`[On/Off]`

Print a message if an optional actual parameter is omitted.

Default is on.

Set-Warn-Output

Parameters:

`[On/Off]`

Print a warning message if an SDL signal output has a different semantic meaning between SDL 88 and 92.

Default is on.

Note:

This command is particularly useful when working with SDT 2.X diagrams in an SDT 3.X environment (SDT 2.X supports SDL-88, while SDT 3.X supports SDL-92).

Set-Warn-Parameter-Count

Parameters:

`[On/Off]`

Print a message if trailing actual parameters are omitted.

Default is on.

Set-Warn-Usage

Parameters:

`[On/Off]`

Print a message if an SDL definition is not used.

Default is off.

Set-Xref

Parameters:

`[On/Off]`

With this option on, a file containing *SDL Cross References* will be generated. See "SDL Cross-References" on page 2438.

Default is off.

Show-Analyze-Options

Parameters:

`(None)`

This command lists the current Analyzer options.

Show-Commands

Parameters:

`(None)`

Similar to *Help*. The commands are listed without the command parameters.

Show-Generate-Options

Parameters:

```
(None)
```

This command displays the code generator options as currently defined.

Show-License

Parameters:

```
(None)
```

This command returns information about:

- The total amount of software licenses
- How many licenses are currently in use, together with the user identification, the host name and the checkout time
- How many licenses are available

Show-Version

Parameters:

```
(None)
```

This command displays the version number of the Analyzer.

Source-Directory

Parameters:

```
[<directory>]
```

The analyzer looks for include files in this directory.

Default is current working directory.

Synonym-File

Parameters:

```
[<file spec>]
```

Specifies the synonym file (containing values of external synonyms). Default is no synonym file.

TAEX-Make-File

Parameters:

```
[<file spec>]
```

Name of makefile for Targeting Expert. See *Set-TAEX-Make*.

Target-Directory

Parameters:

```
[<directory>]
```

Specifies the directory where the code generators puts their files.

Default is current working directory.

Thread

Parameters:

```
<name> <stacksize> <stackprio> <maxqueuesize>
<maxmessize>
```

Use this command to distribute components of a program over several threads. Components following this command belongs to this thread. Parameters at the end may be omitted. Omitted parameters will use default values. The word default can be used to indicate that the default value is desired for a parameter. For more information see "Threaded Integration" on page 3225 in chapter 65, *Integration with Operating Systems*.

XRef-File

Parameters:

```
[<file spec>]
```

Specifies where the cross references are stored.

Default is `<infile>.xrf`.

Miscellaneous Analyzer Commands

The following commands can be used for instance in the case of malfunction or unexpected behavior. They are not described further within the scope of the user documentation.

zzMake-Options

Parameters:

```
(None)
```

zzSet-Access-Path-Tab

Parameters:

```
(None)
```

zzSet-Optimize
Parameters:
(None)

zzSet-Organizer-File
Parameters:
(None)

zzSet-Over-View
Parameters:
(None)

zzSet-SDT
Parameters:
(None)

zzSet-Verbose
Parameters:
(None)

zzSet-Warn-Oper-Usage
Parameters:
(None)

zzShow-Error
Parameters:
(None)

zzShow-Organizer
Parameters:
(None)

zzTransport
Parameters:
(None)

zzWrite-Name-List
Parameters:
(None)

zzWrite-Path
Parameters:
(None)

zzWrite-Pretty
Parameters:
(None)

zzWrite-Symbol
Parameters:
(None)

zzWrite-Syntax
Parameters:
(None)

Conversion to PR

During the first pass of the Analyzer, an SDL/PR file suitable for processing by the following passes (Macro Expansion or Syntax analysis) is produced. The resulting PR file is not intended to be read by the human, and is not formatted for that purpose. Conversion to PR is needed when input is not already in SDL/PR format, i.e, for SDL/GR diagrams, ASN.1 files, and C header files.

Several tools might be invoked through the PostMaster to perform the actual conversion:

- The SDL Editor to process SDL/GR. See "GR to PR Conversion" on page 1981 in chapter 44, *Using the SDL Editor*.
- The ASN.1 Utilities to process ASN.1. See "Translation of ASN.1 to SDL" on page 700 in chapter 14, *The ASN.1 Utilities*.
- The CPP2SDL Utility to process C/C++ header files. See "C/C++ to SDL Translation Rules" on page 778 in chapter 15, *The CPP2SDL Tool*.

Additional SDL/PR files may be copied into the output of this pass.

A number of checks are performed during this conversion pass. Many of these are actually syntactic in their nature. They may result in error messages; see "Error and Warning Messages" on page 2461.

The Macro Expander

A macro is a form of text substitution used in the SDL system definitions. In a macro definition, a collection of lexical units is defined. A macro call indicates where the text from the macro definition body is to be included. All macro calls must be expanded before further analysis of the system definition can be done.

> **Note:**
>
> **Macros may not contain any states.**
>
> **Diagrams referenced in PR form and macros only called in PR will not be converted. If a macro is called only in PR form, the macro definition must be in PR and placed in a diagram that will be converted.**

The name of the macro definition is visible in the whole system definition. A macro call may appear before its corresponding macro definition. A macro definition may contain macro calls, but not calls to itself, directly or indirectly.

A macro may use parameters, called formal parameters, in macro definitions and actual parameters in macro calls. There must be an equal number of formal parameters and actual parameters in corresponding macro definitions and macro calls. Using multiple formal parameters with the same name is not allowed.

When a character string is used as an actual parameter, the value of the character string will be used to replace the formal parameter in the macro body and not the character string itself.

The keyword *MACROID* may be used in a macro definition body only. The *MACROID* keyword will be replaced by a unique name at each expansion of the macro (that is, only one unique name is available at each expansion of a macro definition). New names can be constructed by concatenating names, parameters and *MACROID* using a percent sign (%) as a separator.

The reserved word *MACRODEFINITION* is not allowed inside a macro definition. The reserved words *MACROID* and *ENDMACRO* are only allowed inside a macro definition.

Macro expansion follows this order:

1. The name in the macro call is used to find the corresponding macro definition, and a copy of that macro definition is made.
2. In this copy, all occurrences of formal parameters are replaced by actual parameters. All occurrences of *MACROID* are replaced by a unique name. Also, all percent signs (%) separating names are removed.
3. Macro calls in the modified macro definition body are expanded.
4. The modified and expanded text is substituted for the macro call text in the system definition.

Implementation Details

The syntax checks made in the macro parser are entirely specific to macros since no other checks are possible until all macro calls are expanded. Some semantic checks are made during the expansion. These may produce error messages; see "Error and Warning Messages" on page 2461.

Example 329: Macro in SDL that Will Not Work

Constructs such as:

```
macro test('if par3 then',
           'if not par3 then',
           'k<0') ;
```

where the text `par3` in the first and second parameter strings is intended to be replaced by the third formal parameter in the macro definition will not work. This is because all formal parameters are replaced with actual parameters in a single step.

The reserved word *MACROID* is replaced by a unique name consisting of the string *"XMID"* and an integer. To assure system wide uniqueness the name is tested against all existing names in the system and the integer will be increased until the name is unique.

The Lexical and Syntactic Analyzer

The scanner (*lexical analyzer*) reads the input file and passes on lexical units to the parser. The parser checks the syntax and builds an internal representation (an abstract syntax tree) for the SDL specification.

The lexical and syntactic analysis may be performed not only on complete system definitions, but also on the following SDL units:

- *Block*
- *Process*
- *Substructure*
- *Service*
- *Procedure*
- *Packag*e

If no errors are found during the syntactic analysis, the syntax tree can be passed to the pretty-printer and, if the input is a system or package definition, to the semantic analyzer. For a description of the errors that can produced, see "Error and Warning Messages" on page 2461.

Separate Analysis

You may perform analysis on separate SDL units. The units that can be handled (other than system) are:

- *Block*
- *Process*
- *Substructure*
- *Service*
- *Procedure*
- *Package*

Note:

When performing separate analysis on a package, referred packages must be supplied as well.

If a unit is given without context, only syntactic checking can be performed. But, if the unit's placement in the system (that is, its context) is also given, both the syntactic checking and most of the semantic checking can be performed. The context of a unit is its enclosing scope units including the definitions that are not referenced, i.e. remote diagrams are excluded.

GR Input

In the Analyzer, the context is specified in the Organizer tree structure if the input is a GR diagram. The diagrams surrounding the diagram that is to be separately analyzed and the diagrams inside that diagram are translated to PR and analyzed. No errors will occur due to missing remote diagrams, except missing remote diagrams inside the separately analyzed diagram. See "Performing Syntax Check" on page 2550 in chapter 56, *Analyzing a System*.

PR Input

If the input is a PR file, there is no way to specify what part of the system to Analyze. Only syntax analysis is available on incomplete systems in PR form. See "Converting SDL/PR to SDL/GR" on page 2558 in chapter 56, *Analyzing a System*.

Including PR Files

The Analyzer allows the user to divide the SDL description into a number of separate PR files. (The macro expansion can, however, only handle one file at a time.) It may, for example, be convenient to have the system level data type definitions in a separate file. A separate file is included by adding an *include directive* to the SDL description.

Syntax of #INCLUDE Directives

The include directive is `#INCLUDE` followed by the name of the file which should be surrounded by single quotation marks. The directive should be placed in an SDL comment, directly after the comment start ("/*"), at the place where the file should be included.

- The first single quote must be preceded by at least one space. `<TAB>` characters are however not allowed.
- The second single quote may be followed by any number of spaces. `<TAB>` characters are however not allowed.

Example 330: #INCLUDE in Analyzer ––––––––––––––––––––––––

```
System Example;
/*#INCLUDE 'DataDefs.pr' */
/*#INCLUDE 'BlockA.pr' */
EndSystem Example;
```

––

Search Order for Included PR Files

The Analyzer will search for included SDL/PR files in the following order:

1. Any directory specified with Include-Directory command
2. Source directory
3. The current default directory
4. The directory designated by the environment variable `HOME`.
5. The directory designated by `$telelogic/sdt/include/ADT` **(on UNIX)**, or <*installation directory*>`\sdt\include\adt` **(in Windows)**
   (This directory contains a number of useful abstract data types in

SDL/PR that are included in the release. See chapter 63, *The ADT Library* for more information.)

The PR to GR Converter

General

The PR to GR Converter performs a conversion from SDL/PR files into SDL/GR diagrams. (It is also possible to input SDL/GR diagrams, which transforms them to new SDL/GR diagrams, applying default layouting algorithms.)

Conversion is done on a one-diagram-per-file basis, meaning that each generated GR diagram will be stored on one file. However, entering a PR file may generate multiple GR diagrams, depending on the contents of the PR file.

Note:

The PR to GR converter requires **syntactically correct** diagrams or PR files as input in order to function properly.

Diagrams containing semantic errors will be converted, but, the **resulting** SDL/GR interpretation may be different from the SDL/PR representation.

Conversion Principles

The Analyzer passes which are performed are:

1. Macro expansion (if the PR input contains macro definitions).
2. Syntax analysis.
3. PR to GR conversion.
4. Then, graphical layouting is done by the SDL Editor.

Resulting files

The result from the PR to GR conversion is a number of SDL/GR binary files. Each diagram will be stored on one file.

To reconstruct the SDL diagram structure, the Organizer's command *Import SDL* should be used (see "Import SDL" on page 79 in chapter 2, *The Organizer*).

SDL Cross-References

The Analyzer has the ability to produce listings of SDL entities and references to these definitions. In one file, all *definitions* of entities in an SDL system will be gathered, along with cross references to these entities. Entities mainly used to indicate structure (such as system, block, substructure, process, procedure, and service), as well as entities defining objects (such as signals, variables and sorts) will be considered as definitions.

The file, which is described further below, is a plain text file.

> **Note:**
>
> The SDL suite has support for graphical presentation of a cross-reference file, using an SDL-like graphical notation. To achieve this, you can use the Index Viewer. See chapter 47, *The SDL Index Viewer*.

Definitions and Cross References Files

The file name will be `<unitname>.xrf` for the *cross references file*, where <unitname> is the name of the SDL unit selected for analysis. Alternatively, the file name is to be supplied by the user when running the Analyzer from the Organizer.

If an SDL system is selected for analysis, all definitions and cross references are generated; while if case of a non-SDL system unit, definitions and cross references in the following units are generated:

- The unit itself
- All subunits to the selected unit
- All enclosing units (blocks and the system) of the selected unit

The following entities will be part of the files:

ACTIVE	ATLEAST

BLOCK	BLOCK SUBSTRUCTURE
CHANNEL	CHANNEL SUBSTRUCTURE
CONNECTION	CONTINUOUS SIGNAL
CREATE	DCL
DECISION	ENABLING CONDITION
EXPORT	EXPORTED_PROCEDURE
FORMAL PARAMETER	FPAR
GATE	GENERATOR
IMPORT	IMPORTED
IMPORTED VARIABLE	IN-CONNECTOR
INHERITS	INPUT
INSTANTIATION	JOIN
LITERAL	NEWTYPE
NEXTSTATE	NUMBER OF INSTANCES
OPERATOR	OUT-CONNECTOR
OUTPUT	PACKAGE_INTERFACE
PROCEDURE	PROCEDURE CALL
PROCEDURE PARAMETERS	PROCESS
PROCESS PARAMETERS	REMOTE PROCEDURE
REMOTE VARIABLE	RESET
SAVE	SERVICE
SET	SIGNAL
SIGNAL ROUTE	SIGNALLIST
SIGNALROUTE	SIGNALSET
SORT	STATE
STATE_LIST	SYNONYM
SYNTYPE	SYSTEM

TASK	TIMER
TRANSITION OPTION	USE
VARIABLE	VIEW
VIEWED	

Note:

No cross references will be generated for the predefined data types (INTEGER, NATURAL, CHARACTER, CHARSTRING, BOOLEAN, REAL, TIME, DURATION, PID) or for the predefined generators (ARRAY, STRING, POWERSET) as well as LITERALS and OPERATORS that are not part of the list above.

Syntax of Files

For each definition of any kind mentioned above the following information is generated:

```
ScopeLevel EntityType EntityName Reference
```

Explanation

- The number first on the line is the *scope level*. The system has level 1, any definition at the system level has scope level 2, and so on.
- Next on the line the *entity type* and *entity name* are given.
- Last there is an *SDT reference* to the place where the entity is defined. For more information on the format, see "Syntax" on page 911 in chapter 19, *SDT References*.

For each entity used to indicate structure (system, block, substructure, process, procedure, service) there is a header consisting of three additional lines, which should be considered as a comment to increase the readability.

Example 331: Reference in Analyzer Error

```
2  SIGNAL Bump \
SDTREF(SDL,/usr/tom/demongame.ssy(1),122(40,30),3)
```

Order of Definitions

The definitions are generated in pre-order (prefix walk in the tree formed by the definitions). This means that an entity is always followed by the entities defined within the entity. It also means that an entity at level N is defined in the first entity at level N-1 found when scanning upwards in the file.

Cross References

For each place where an entity is used, information about that cross reference is generated:

Example 332: Cross References ––––––––––––––––––––––––––––

```
4  DCL Count \
#SDTREF(SDL,/usr/tom/game.spr(1),179(80,10),2)
     TASK \
#SDTREF(SDL,/usr/tom/game.spr(1),137(30,40),1)
     OUTPUT \
#SDTREF(SDL,/usr/tom/game.spr(1),128(80,55),2)
```

––

Cross references consist of a keyword, describing where the reference was found, and the SDL reference. In Example 332 on page 2441, the variable Count is referenced in a task symbol and in an output. All cross references for an entity are placed immediately after the line for the definition.

Note:

Entities used to indicate structure can also have cross references (procedure call, process create).

SDL Instance Information

The Analyzer supports the generation of a so-called instance information file, with the file extension `.ins`. This file contains various kinds of information about an SDL system. Similar to a cross reference file, it is a plain text file. The file syntax is described in detail in "File Syntax" on page 2443. The Analyzer invokes a tool called the Instance Generator to produce the instance information out of an analyzed SDL system.

The information that can be found in an instance information file helps to answer many questions. For example:

- How does a complex SDL-92 system look after instantiation of all types, i.e. when all inheritance hierarchies have been flattened out? Finding this information about the instance structure of a system directly from the SDL/PR file is a non-trivial task. Therefore, one of the main objectives behind the Instance Generator is to provide this "instance view" of a system.
- Which signals can be conveyed on a certain channel or signal route in a certain direction?
- What is the valid input signal set of a process?
- Which states does a process contain?
- Which transitions will take place if a certain signal is received by a process being in a certain state?
- Which output signals might be sent during a transition?
- Which procedures might get called during a transition?
- What is the set of possible nextstates after a transition?
- How shall a procedure be interpreted when called from a certain process?

The Instance Generator

The Analyzer provides two commands to use the Instance Generator from the command line user interface. These commands are described in "The Analyzer Command-Line UI" on page 2404 but are repeated here for the sake of completeness.

Set-Instance

Parameters:

`[On/Off]`

This command sets an option that specifies whether an instance information file should be generated or not when the SDL system is analyzed. The option is by default off.

Instance-File

Parameters:

`<file spec>`

This command sets the name of the instance information file. The default filename is `<systemname>.ins`, where `<systemname>` is the name of the analyzed SDL system.

The Instance Generator can also be started from the Analyzer's graphical user interface. This is done in a way similar to how cross reference files are generated. In the Analyze dialog in the Organizer, you can check a button to make the Analyzer call the Instance Generator when the system has been analyzed. From this dialog it is also possible to set the name of the generated instance information file.

Note:

The Instance Generator is automatically invoked to produce the information needed by tools such as the State Matrix Viewer.

File Syntax

An instance information file consists of a *fileheader* followed by a list of *records*:

```
<fileheader>
<record>
<record>
...
<record>
```

The `<fileheader>` is a string telling which version of the Instance Generator that has generated the file. Each `<record>` describes an SDL entity according to the following format:

```
<level> <entity class> <name>
        <attribute> = <value>
        ...
        <attribute> = <value>
        .
```

The <level> is a number that tells at what depth in the "instance tree" the entity was found. For example, the system entity has level 1, a block entity in the system has level 2, a process entity in the block has level 3, and so on.

Attributes can be divided into three distinct **categories** depending on the format of their values:

1. A single value:

```
<attribute> = <single value>
```

2. A list of single values:

```
<attribute> = (
  <single value>
  <single value>
  ...
  <single value>
)
```

3. A list of pairs of single values:

```
<attribute> = (
  <single value> <single value
  <single value> <single value>
  ...
  <single value> <single value>
)
```

Sometimes an attribute is omitted from a record. This happens when the value of the attribute is an empty string or list.

The table below lists all entity classes for which information is generated in an instance information file. The attributes of the entities are also described, together with the category of the attributes (1, 2 or 3 according to above).

In the descriptions, the following terms are used:

- A *state-containing entity* is an entity that may contain states, i.e. a process, process instance, service, service instance or procedure.
- A *transition initiator* is an entity that can make a state-containing entity perform a transition from one state to another, i.e. a start, input, priority input, enabling condition or continuous signal.

Entity class	Attribute	Category	Description
BLOCK	InstRef	1	An SDT reference to the definition of this block.
	ConnectionOut	2	The names of the channels that are on the outside of this block.
	ConnectionIn	2	The names of the channels or signal-routes that are on the inside of this block.
BLOCK_INST	InstRef	1	An SDT reference to the definition of this block instance.
	TypeRef	2	SDT references to block type definitions. The first reference is to the block type from which this block instance is instantiated, followed by references to its supertypes all the way to the block type on top of the inheritance hierarchy.
CALLED_ PROCEDURE	InstRef	1	An SDT reference to the definition of this called procedure.
CHANNEL	InstRef	1	An SDT reference to the definition of this channel.
	SignalSet	3	The signals that are conveyed on this channel in the direction from the From-entity to the To-entity. Each signal is described by its name followed by an SDT reference to the definition of the signal.
	SignalSetRev	3	As SignalSet but for the signals that are conveyed in the opposite direction.

Entity class	Attribute	Category	Description
	From	1	The name of the block or block instance from which this channel starts. If the channel starts from the environment surrounding the system, system instance, block or block instance where this channel is located, this attribute is “env”.
	FromVia	1	This attribute is specified when the From-entity is a block instance, and is then the name of the gate of that block instance to which this channel is connected.
	To	1	The name of the block or block instance to which this channel leads. If the channel leads to the environment surrounding the system, system instance, block or block instance where this channel is located, this attribute is “env”.
	ToVia	1	This attribute is specified when the To-entity is a block instance, and is then the name of the gate of that block instance to which this channel is connected.
CHANNEL_SUBSTRUCTURE	InstRef	1	An SDT reference to the definition of this channel substructure.
CONTINUOUS_SIGNAL[a]	InstRef	1	An SDT reference to the definition of this continuous signal.
	InType	1	The name of the state-containing entity in which this continuous signal is defined.

Entity class	Attribute	Category	Description
	Outputs	3	The output signals that might be sent in the transition that is initiated by this continuous signal. Each signal is described by its name followed by an SDT reference to one of the output symbols in the transition that contains that name.
	Calls	3	The procedures that might get called in the transition that is initiated by this continuous signal. Each procedure is described by its name followed by an SDT reference to one of the symbols in the transition that contains a call to a procedure with that name. This name is also the name of a CALLED_PROCEDURE entity that describes how the procedure look from the calling state-containing entity's point of view. This entity is normally placed immediately after the state information of the state-containing entity. However, when the state-containing entity is a procedure within another state-containing entity, it might be that the procedure has been called from another place in this surrounding state-containing entity. Then the CALLED_PROCEDURE entity is not placed here also, since it is identical to the first one.
	NextStates	3	The states that might become the next-state after the transition that is initiated by this continuous signal. Each state is described by a name followed by an SDT reference to one of the state symbols that contains that name.

Entity class	Attribute	Category	Description
ENABLING_CONDITION	InstRef	1	See the description for CONTINUOUS_SIGNAL.
	InType	1	See the description for CONTINUOUS_SIGNAL.
	Outputs	3	See the description for CONTINUOUS_SIGNAL.
	Calls	3	See the description for CONTINUOUS_SIGNAL.
	NextStates	3	See the description for CONTINUOUS_SIGNAL.
GATE	InstRef	1	An SDT reference to the definition of this gate.
	SignalSet	3	The signals that are conveyed through this gate in to the block instance, process instance or service instance to which the gate belongs. Each signal is described by its name followed by an SDT reference to the definition of the signal.
	SignalSetRev	3	As SignalSet but for the signals that are conveyed in the opposite direction.
INPUT	InstRef	1	See the description for CONTINUOUS_SIGNAL.
	InType	1	See the description for CONTINUOUS_SIGNAL.
	Outputs	3	See the description for CONTINUOUS_SIGNAL.
	Calls	3	See the description for CONTINUOUS_SIGNAL.
	NextStates	3	See the description for CONTINUOUS_SIGNAL.

Entity class	Attribute	Category	Description
PRIORITY_INPUT	InstRef	1	See the description for CONTINUOUS_SIGNAL.
	InType	1	See the description for CONTINUOUS_SIGNAL.
	Outputs	3	See the description for CONTINUOUS_SIGNAL.
	Calls	3	See the description for CONTINUOUS_SIGNAL.
	NextStates	3	See the description for CONTINUOUS_SIGNAL.
PROCESS	InstRef	1	An SDT reference to the definition of this process.
	IniNoOfInst	1	The initial number of dynamic instances of this process.
	MaxNoOfInst	1	The maximum number of dynamic instances of this process. If this attribute is omitted it means that there is no upper limit on the number of dynamic instances.
	SignalSet	3	The signals that can be received by this process. This set of signals is commonly known as the valid input signal set of the process. Each signal is described by its name followed by an SDT reference to the definition of the signal.
	ConnectionOut	2	The names of the signalroutes that are on the outside of this process. Naturally, this attribute is only present when the process consists of services.

Entity class	Attribute	Category	Description
	ConnectionIn	2	The names of the signalroutes that are on the inside of this process. Naturally, this attribute is only present when the process consists of services.
PROCESS_INST	InstRef	1	An SDT reference to the definition of this process instance.
	TypeRef	2	SDT references to process type definitions. The first reference is to the process type from which this process instance is instantiated followed by references to its supertypes all the way to the process type on top of the inheritance hierarchy.
	IniNoOfInst	1	The initial number of dynamic instances of this process instance.
	MaxNoOfInst	1	The maximum number of dynamic instances of this process instance. If this attribute is omitted it means that there is no upper limit on the number of dynamic instances.
	SignalSet	3	The signals that can be received by this process instance. This set of signals is commonly known as the valid input signal set of the process instance. Each signal is described by its name followed by an SDT reference to the definition of the signal.
SAVE	InstRef	1	An SDT reference to the definition of this save.
	InType	1	The name of the state-containing entity in which this save is defined.
SERVICE	InstRef	1	An SDT reference to the definition of this service.

Entity class	Attribute	Category	Description
SERVICE_INST	InstRef	1	An SDT reference to the definition of this service instance.
	TypeRef	2	SDT references to service type definitions. The first reference is to the service type from which this service instance is instantiated followed by references to its supertypes all the way to the service type on top of the inheritance hierarchy.
SIGNALROUTE	InstRef	1	An SDT reference to the definition of this signalroute.
	SignalSet	3	The signals that are conveyed on this signalroute in the direction from the From-entity to the To-entity. Each signal is described by its name followed by an SDT reference to the definition of the signal.
	SignalSetRev	3	As SignalSet but for the signals that are conveyed in the opposite direction.
	From	1	The name of the process, process instance, service or service instance from which this signalroute starts. If the signalroute starts from the environment surrounding the block, block instance, process or process instance where this signalroute is located, this attribute is “env”.
	FromVia	1	This attribute is specified when the From-entity is a process instance or a service instance, and is then the name of the gate of that process instance or service instance to which this signalroute is connected.

Entity class	Attribute	Category	Description
	To	1	The name of the process, process instance, service or service instance to which this signalroute leads. If the signalroute leads to the environment surrounding the block, block instance, process or process instance where this signalroute is located, this attribute is "env".
	ToVia	1	This attribute is specified when the To-entity is a process instance or a service instance, and is then the name of the gate of that process instance or service instance to which this signalroute is connected.
START[b]	InstRef	1	See the description for CONTINUOUS_SIGNAL.
	InType	1	See the description for CONTINUOUS_SIGNAL.
	Outputs	3	See the description for CONTINUOUS_SIGNAL.
	Calls	3	See the description for CONTINUOUS_SIGNAL.
	NextStates	3	See the description for CONTINUOUS_SIGNAL.
STATE	InstRef	2	SDT references to the state symbols in the state-containing entity that contain the name of this state. If the state-containing entity is a process type, service type or procedure, occurrences in the supertypes are also included.
SUBSTRUCTURE	ConnectionOut	2	The names of the channels that are on the outside of this substructure.

Entity class	Attribute	Category	Description
	ConnectionIn	2	The names of the channels or signalroutes that are on the inside of this substructure.
SYSTEM	InstRef	1	An SDT reference to the definition of this system.
SYSTEM_INST	InstRef	1	An SDT reference to the definition of this system instance.
	TypeRef	2	SDT references to system type definitions. The first reference is to the system type from which this system instance is instantiated followed by references to its supertypes all the way to the system type on top of the inheritance hierarchy.

a. The name of a CONTINUOUS_SIGNAL entity is set to the priority of the continuous signal. If no priority has been specified, maximum priority is assumed and the name is then set to "MAX_PRIO".

b. The name of a START entity is always set to "Start".

Information about the entities is generated in pre-order (prefix walk in the instance tree). This means that an entity is always followed by the entities defined within itself.

A formal and complete description of the general format of an instance information file would be rather complex, and is beyond the scope of this text. Instead, see Example 333:

Example 333: An instance information file

Consider the SDL system below:

```
System ExSystem;
  Signal Sig(Integer), Inp;
  channel ExChannel
    from ExBlock to env with Sig;
    from env to ExBlock with Inp;
  endchannel ExChannel;
  block ExBlock referenced;
endsystem ExSystem;

Block ExBlock;
  signalroute ExSignalRoute
    from ExProcess to env with Sig;
    from env to ExProcess with Inp;
  process ExProcess referenced;
  connect ExChannel and ExSignalRoute;
endblock ExBlock;

Process ExProcess;
  DCL Counter Integer;
  procedure ExProcedure referenced;
  start ;
    task Counter := 0;
    grst4:
  nextstate Wait;
  state Wait;
    input Inp;
    task {
      Counter := call ExProcedure(Counter);
    };
    output Sig(Counter);
    join grst4;
  endstate;
endprocess ExProcess;

Procedure ExProcedure;FPAR a Integer;RETURNS ret
Integer;
  start ;
  nextstate Idle;
  state Idle;
    input *;
    task {
      ret := a+1;
    };
    return ;
  endstate;
endprocedure ExProcedure;
```

The instance information file for this SDL system will look like this:

```
SDTINST V1.0
1  SYSTEM ExSystem
   InstRef = #SDTREF(TEXT,ExSystem.pr,1)
   .
2  CHANNEL ExChannel
   InstRef = #SDTREF(TEXT,ExSystem.pr,3)
   SignalSet = (
     Sig #SDTREF(TEXT,ExSystem.pr,2)
   )
   SignalSetRev = (
     Inp #SDTREF(TEXT,ExSystem.pr,2)
   )
   From = ExBlock
   To = env
   .
2  BLOCK ExBlock
   InstRef = #SDTREF(TEXT,ExSystem.pr,10)
   ConnectionOut = (
     ExChannel
   )
   ConnectionIn = (
     ExSignalRoute
   )
   .
3  SIGNALROUTE ExSignalRoute
   InstRef = #SDTREF(TEXT,ExSystem.pr,11)
   SignalSet = (
     Sig #SDTREF(TEXT,ExSystem.pr,2)
   )
   SignalSetRev = (
     Inp #SDTREF(TEXT,ExSystem.pr,2)
   )
   From = ExProcess
   To = env
   .
3  PROCESS ExProcess
   InstRef = #SDTREF(TEXT,ExSystem.pr,18)
   IniNoOfInst = 1
   SignalSet = (
     Inp #SDTREF(TEXT,ExSystem.pr,2)
   )
   .
4  START Start
   InstRef = #SDTREF(TEXT,ExSystem.pr,21)
   InType = ExProcess
   Nextstates = (
     Wait #SDTREF(TEXT,ExSystem.pr,24)
   )
   .
4  STATE Wait
   InstRef = (
     #SDTREF(TEXT,ExSystem.pr,25)
   )
   .
```

```
5  INPUT Inp
   InstRef = #SDTREF(TEXT,ExSystem.pr,26)
   InType = ExProcess
   Outputs = (
     Sig #SDTREF(TEXT,ExSystem.pr,30)
   )
   Calls = (
     ExProcedure #SDTREF(TEXT,ExSystem.pr,28)
   )
   Nextstates = (
     Wait #SDTREF(TEXT,ExSystem.pr,24)
   )
   .
4  CALLED_PROCEDURE ExProcedure
   InstRef = #SDTREF(TEXT,ExSystem.pr,20)
   .
5  START Start
   InstRef = #SDTREF(TEXT,ExSystem.pr,36)
   InType = ExProcedure
   Nextstates = (
     Idle #SDTREF(TEXT,ExSystem.pr,37)
   )
   .
5  STATE Idle
   InstRef = (
     #SDTREF(TEXT,ExSystem.pr,38)
   )
   .
6  INPUT Inp
   InstRef = #SDTREF(TEXT,ExSystem.pr,39)
   InType = ExProcedure
   Nextstates = (
     return #SDTREF(TEXT,ExSystem.pr,43)
   )
   .
```

Error Handling

Errors may be detected both during interaction with the user (command interpretation) and during analysis. Errors are displayed on the screen and optionally appended to an error file supplied by the user. When using the graphical UI, messages are directed to the Organizer log window.

Command Interpretation Errors

Errors that occur during command interpretation are errors due to:

- Illegal command parameters. See "Syntax of Analyzer Commands" on page 2405.
- Problems with accessing files on your computer system.

Diagnostics Issued During Analysis

Errors may be found during the various steps of the analysis. The different types of diagnostics and the format of the messages are described below. The messages are listed in "Error and Warning Messages" on page 2461 in chapter 55, *The SDL Analyzer*.

Diagnostic Types

Errors can be of the following types: warning, error, internal error, and information.

Warning

There are different types of warnings:

- Warnings that truncation has been performed, because the implementation limits have been exceeded.
- Warnings due to non implemented portions of the Analyzer.
- Warnings due to a badly designed SDL construction that is not essential for the interpretation.
- Warnings due to analysis actions that were not performed because the Analyzer could not recover from errors in previous analysis steps.

Error

One type of error is detected; violation of an SDL rule.

Internal Error

Internal errors due to malfunction of the Analyzer are detected. An internal error often has its roots in an SDL error from which the Analyzer did not recover properly.

Information

Often used to provide additional information to the other types of messages.

Diagnostic Format

Messages generally consist of the following parts:

- A reference to the source code that caused the diagnostic to be reported. See "Syntax" on page 911 in chapter 19, *SDT References*.
- The type of the diagnostic (ERROR / WARNING / INFO).
- A number identifying the diagnostic.
- A message describing the diagnostic.

Example of a Syntax Error

The message describing the error includes a text line where the illegal construct is indicated with a "?". If the trace back is common to many lexical or syntactic errors, the trace back will only occur after the last of these error messages.

The error also contains a reference to the source file.

Example 334: Syntax Error Produced by Analyzer ––––––––––––––––

```
#SDTREF(SDL,/usr/tom/game.spr(1),137(30,40),1,11)
ERROR 312 Syntax error in rule VARIABLE, symbol =
found but one of the following
expected:
!  (  :=
task Count=0 ;
         ?
```

––

The interpretation of the error message is that the Analyzer found the symbol "=" when it expected one of the symbols "!" (the field selector), "(" or ":=".

In the current example, the assignment statement can be found in position 11 the first line in the object with the ID 137 at position (30, 40) on page 3, on page 1 in the diagram stored on file `/usr/tom/game.spr`.

Example of a Semantic Error

The message describing the error includes, if possible, a line containing the error, with a "?" immediately preceding the SDL item that caused the error.

The error also contains a reference to the source file.

Example 335: Semantic Error Produced by Analyzer ––––––––––––––

```
#SDTREF(SDL,/usr/tom/game.spr(1),296(55,40),1)
ERROR 88 Undefined procedure
call ? ErrorHandler
```

––

The interpretation of this error message is that the procedure ErrorHandler is referred to but not is defined, and the call can be found in page 1 of the diagram stored on the on file `/usr/tom/game.spr`, in line 1 of the object with ID 296 and at position (55, 40).

Analyzer Files

When a diagram is being analyzed, a number of files are generated. The file names consist of the diagram name appended with a file name extension. The following file name extensions are used:

- `.pr`
  The SDL/PR, (also known as *phrasal representation*), of the SDL diagram. This file is generated by the conversion to PR pass. The purpose of this file is an intermediate format for the Analyzer. The format used in this file makes it not suitable to be read by the human.

- `.prm`
  After macros are expanded, the `.pr` file is used as input and expanded into a `.prm` file, which includes the SDL macros in an expanded form. The appearance of this file makes it not suitable to be read by the human.

- `.sdl`
  The pretty-printed SDL/PR file which is generated by the pretty-printer. This file uses a layout that is easy to read by the human.

- `.err`
  The error file containing the errors and warnings which were detected during the analysis.

- `.xrf`
  The files containing the listings of SDL definitions and cross-references.

- `.tsp`
  The file contains time stamp information from the last analysis.

- `.ins`
  The file containing instance information about the analyzed SDL system. Read more about the contents of this file in "SDL Instance Information" on page 2442.

- `predef.sdl`
  This file contains predefined SDL entities that the Analyzer needs access to in order to function properly. The file is found using the procedure in "Search Order for Included PR Files" on page 2436 with the directory `$sdtdir`.

Error and Warning Messages

This section contains a list of the Analyzer error and warning messages. Each message has a short explanation and, where applicable, a reference to the appropriate section of the recommendation Z.100, or to the formal definition. Z.100 is appended by the Formal Definition of SDL (Annex F.1-F.3) where a formal mathematical definition of the language is given.

Some messages contain a '%' followed by a number, which is used to indicate where information, specific to the error situation, will be included.

Some messages include a reference to the object that is the source of the diagnostic. These messages adhere to the format adopted in the SDL suite. See chapter 19, *SDT References* for a reference to this format and for examples.

`INFO 0 Internal error: message %1 not found`

This message indicates an error in the implementation of the Analyzer. Please send a report to Telelogic Customer Support, especially if the error can be reproduced as the only error message of an analysis. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

`ERROR 1 Internal error`

This message indicates an error in the implementation of the Analyzer. Please send a report to Telelogic Customer Support, especially if the error can be reproduced as the only error message of an analysis. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

`ERROR 2 Cannot open file: %1`

This error message indicates that an error occurred when the Analyzer attempted to open a file. Modify, if necessary, the file protection and try to run the Analyzer again. If the error persists, contact Telelogic Customer Support. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

ERROR 3 Cannot close file: %1

This error message indicates that an error occurred when the Analyzer attempted to close a file. Modify, if necessary, the file protection and try to run the Analyzer again. If the error persists, contact Telelogic Customer Support. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

ERROR 4 Cannot delete file: %1

This error message indicates that an error occurred when the Analyzer attempted to delete a file. Modify, if necessary, the file protection and try to run the Analyzer again. If the error persists, contact Telelogic Customer Support. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

ERROR 5 Cannot rename file: %1 to %2

This error message indicates that an error occurred when the Analyzer attempted to rename a file. Modify, if necessary, the file protection and try to run the Analyzer again. If the error persists, contact Telelogic Customer Support. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

ERROR 6 Recursive definition of signal list %1

A signal list must not be based on itself, directly or indirectly.
(Z.100: 2.5.5)

ERROR 7 More than one %1 is visible in Packages

Several definitions with same name were found in a package. A qualifier might resolve the problem.
(Z.100: 2.4.1.2)

ERROR 8 Ending identifier (%2) must be equal to defining identifier (%1) in a referenced definition

When writing a definition, the name must be the same on the last row as on the first row.
(Z.100: 2.2.2)

ERROR 9 Referenced definitions are not unique (%1)

There are more than one definition which matches the reference.

You get this error if:

- You use the same diagram name for two different diagram references in one diagram.

To remove the error:

- Rename one of the referenced diagrams.

(Z.100: 2.4.1.3)

ERROR 10 Number of instances must be lexically equal in referenced process definition and process reference

(Z.100: 2.4.4)

ERROR 11 Different virtuality attributes in reference and referenced heading

(Z.100: 6.3.2)

ERROR 12 A referenced definition may not reference itself

It is not allowed to have a reference to itself in the reference definition.
(Z.100: 2.4.1.3)

ERROR 13 No referenced definition matches reference

A reference is found but not the corresponding definition.

You get this error if:

- You define a diagram as REFERENCED, but do not give a non-referenced definition of the diagram later.

To remove the error, either:

- Add a non-referenced diagram definition.
- Remove the diagram reference.

(Z.100: 2.4.1.3)

ERROR 14 Virtual start not allowed in operator definition

It is not allowed to have the start as virtual in an operator definition.

You get this error if:

- You use the keyword VIRTUAL for the start transition in an operator.

To remove the error:

- Remove the VIRTUAL keyword from the start transition.

(Z.100: 5.3.2)

INFO 15 Required item %1 is not connected to a file

The Organizer has ordered to analyze something depending on a diagram that is not connected to a file.

ERROR 16 Input %1 and output %2 should be different, aborting!

An Analyzer pass requires the input and output files to be different. Specify other input and/or output files.

ERROR 17 Definition name may not be qualified when not a referenced definition

A qualifier is only allowed on a definition if it is a referenced definition.

You get this error if:

- You use a qualifier when defining a diagram in place, i.e. without using the REFERENCED keyword.

To remove the error, either:

- Remove the qualifier.
- Use the keyword REFERENCED and move the definition of the child diagram out of the parent diagram.

(Z.100: 2.4.1.3)

ERROR 18 Referenced definition (%1) is not referenced in the Package definition

A reference to the definition (%1) is missing in the package definition. (Z.100: 2.4.1.3)

ERROR 19 Referenced definition (%1) is not referenced in the system definition

A reference to the definition (%1) is missing in the system definition.

You get this error if:

- You define a diagram outside the SDL system but do not reference it from the SDL system.

To remove the error, either:

- Introduce a reference construct in the SDL system to the diagram.
- Remove the unwanted diagram.

(Z.100: 2.4.1.3)

ERROR 20 Invalid separate analysis unit

It is not allowed to make a separate analysis of the unit selected. Start the analysis on the whole specification instead.

ERROR 21 Referenced substructure definition without name

It is not allowed to have a substructure without a name when referencing it or converting it to SDL/GR.
(Z.100: 3.2.2, 3.2.3)

ERROR 22 System definition may not occur as referenced definition to a system

Only system types may be referenced, not systems.
(Z.100: 2.4.1.3)

ERROR 23 Illegally placed process context parameter

A process context parameter is used where it is not allowed. Examples of bad use is in a system type or in a block type.
(Z.100: 6.2, 6.1.1.1, 6.1.1.2)

ERROR 24 Illegally placed procedure context parameter

A procedure context parameter is used where it is not allowed.
(Z.100: 6.2)

ERROR 25 Illegally placed signal context parameter

A signal context parameter is used where it is not allowed.
(Z.100: 6.2)

ERROR 26 Illegally placed timer context parameter

A timer context parameter is used where it is not allowed. For example in a system type, block type or service type.
(Z.100: 6.2, 6.1.1.1, 6.1.1.2, 6.1.1.4)

ERROR 27 Illegally placed variable context parameter

A variable context parameter is used where it is not allowed. For example in a system type or block type.
(Z.100: 6.2, 6.1.1.1, 6.1.1.2)

ERROR 28 Illegally placed synonym context parameter

A synonym context parameter is used where it is not allowed. For example in a signal.
(Z.100: 6.2, 2.5.4)

ERROR 29 Illegally placed remote procedure context parameter

A remote procedure context parameter is used where it is not allowed.
(Z.100: 6.2)

ERROR 30 Illegally placed remote variable context parameter

A remote variable context parameter is used where it is not allowed.
(Z.100: 6.2)

ERROR 31 Illegally placed sort context parameter

A sort context parameter is used where it is not allowed.
(Z.100: 6.2)

ERROR 32 Illegal redeclaration of name

All definitions in the same scope unit belonging to the same entity class must have different names, with some exceptions.

You get this error if:

- You use the same name as another item when declaring a new item.

To remove the error, either:

- Change one of the names in all appropriate places.
- Remove one of the declarations if the intention was to use the same variable.

(Z.100: 2.2.2)

ERROR 33 Illegally placed gate definition

A gate is defined where it is not allowed.
(Z.100: 6.1.4)

ERROR 34 %1 is not a time stamp file, will not save time stamps

On successful analysis the Analyzer saves info about this. Examine the file and rename or delete it to allow the Analyzer to save time stamps and possibly avoid unnecessary reanalysis in the future.

ERROR 35 Invalid regular interval

The character strings in a nameclass must be of length one and the first one smaller than the second.
(Z.100: 5.3.1.14)

ERROR 36 Integer literal expected

The name in the priority clause of a continuous signal must be an integer literal. This is also true for the name following a regular element in a name class literal.

You get this error if:

- You type something that is not an integer in a place where an integer is expected.

To remove the error:

- Replace the non-integer with an integer.

(Z.100: 4.11, 5.3.1.14)

ERROR 37 Invalid generator parameter %1

The actual generator parameter must match the type of the formal one.
(Z.100: 5.3.1.12.2)

ERROR 38 Illegal number of arguments

The number of actual parameters must be equal to the number of formal parameters when calling a procedure or creating a process. Similar rules apply for output, input and save of signals and timers.

ERROR 39 More than one default assignment; last from generator %1

A sort must not contain more than one default assignment. When instantiating many generators in a sort, only one of them may contain a default assignment.
(Z.100: 5.4.3.3)

ERROR 40 Recursive instantiation of generator %1

A generator definition must not instantiate itself, directly or indirectly.
(Z.100: 5.3.1.12.1)

ERROR 41 Undefined generator

Attempt to instantiate an undefined generator.
(Z.100: 5.3.1.12.2)

ERROR 42 No virtual definition %1 in super of enclosing definition

It is only allowed to redefine or finalize a definition if it is defined as virtual in the inherited type.

You get this error if:

- You are trying to redefine an entity that is not declared as VIRTUAL in the super type.

To remove the error, either:

- Make sure you are trying to redefine the correct entity.
- Make sure that there is a virtual entity to redefine in the super type.

(Z.100: 6.3.2)

ERROR 43 Ordering must not be defined more than once

All operator signatures in a sort must be different (ordering is shorthand for the relational operators).
(Z.100: 5.3.1.8)

ERROR 44 Name of formal parameter of class type must not equal name of generator %1

A generator must not have a formal parameter of class type with the same name as the generator definition.
(Z.100: 5.3.1.12.1)

ERROR 45 Formal generator parameters of classes literal and constant mixed

Literal and constant parameters must not have the same name.
(Z.100: 5.3.1.12.1)

ERROR 46 Only task, decision and transition option allowed in operator definition

You get this error if:

- You try to include signal sending in an operator.

To remove the error:

- Remove the signal sending.

(Z.100: 5.3.2)

ERROR 47 Only Join and Return allowed in operator definition

(Z.100: 5.3.2)

WARNING 48 Ending name (%1) in an asterisk state may lead to an error

The optional state name ending a state may be defined only if the state list is a single state name, in which case the ending state name should be that state name. The state list for an asterisk state is probably more than a single state name.
(Z.100: 2.6.3)

ERROR 49 A state with more than one state name in the state list cannot contain an ending name (%1)

The optional state name ending a state may be defined only if the state list is a single state name, in which case the ending state name should be that state name.
(Z.100: 2.6.3)

ERROR 50 Ending name (%2) must be equal to defining name (%1)

You get this error if:

- You use different names for a definition at the beginning and at the end of the same definition.

To remove the error, either:

- Change the name either at the beginning or at the end of the definition to get the same name in both places.
- Remove the optional name at the end.

(Z.100: 2.2.2, 2.6.3)

ERROR 51 Illegally placed virtuality

Virtual is used in a place where it is not allowed.

You get this error if:

- You use the keyword VIRTUAL for a transition that is not inside a type-based diagram.

To remove the error, either:

- Remove the VIRTUAL keyword from the transition.
- Make the diagram a type-based diagram.

ERROR 52 Illegally placed context parameter

The context parameter is placed where it is not allowed.
(Z.100: 6.2)

ERROR 53 Illegally placed virtuality constraint

The virtuality constraint is placed where it is not allowed.

You get this error if:

- You use the ATLEAST keyword for a type-based diagram that is not placed inside a type-based diagram.

To remove the error, either:

- Remove the ATLEAST construct.
- Move the type-based diagram with the ATLEAST construct to a type-based diagram.
- Make the diagram containing the type-based diagram with the ATLEAST construct a type-based diagram too.

ERROR 54 Illegally placed specialization

The specialization is placed where it is not allowed.

ERROR 55 Illegally placed instantiation

The instantiation is placed where it is not allowed.

ERROR 56 Packages can only contain type definitions

Systems, blocks, processes and services are not allowed, but their corresponding type are.
(Z.100: 2.4.1.2)

ERROR 57 Undefined package %1 in use clause

An used package is not defined. This error may also occur if you have two mutually dependent packages, which is not allowed. If you use ASN.1 modules or C headers, it may also occur if there are no dependency links between these and the SDL system.

You get this error if:

- You try to use a package that has not yet been declared.

To remove the error, either:

- Add a package definition named %1.
- Remove the use of the package.

ERROR 58 Illegally placed exported attribute

The exported attribute is placed where it is not allowed.

ERROR 59 Omitted actual context parameter

(Z.100: 6.2)

ERROR 60 Parameterized type %1 cannot be used as actual context parameter

A type with context parameters is not allowed to be used as an actual context parameter.
(Z.100: 6.2)

ERROR 61 Undefined actual context parameter

The actual context parameter is undefined or is not visible.
(Z.100: 6.2)

ERROR 62 Illegal number of actual context parameters (>)
(Z.100: 6.2)

ERROR 63 Illegal number of actual context parameters (<)
(Z.100: 6.2)

ERROR 64 Illegal number of actual context parameters (0)
(Z.100: 6.2)

ERROR 65 Formal context parameter %1 cannot be used as super type
It is not allowed to specialize a formal context parameter.
(Z.100: 6.2)

ERROR 66 Several matches
Resolution by context found several possible interpretations, use qualifiers to reduce the number of possibilities.
(Z.100: 2.2.2)

ERROR 67 No matching synonym or literal
You get this error if:

- You are using a synonym or literal of the wrong type. For instance, a constant expression of type integer may be required, but you supply a value of type real.

To remove the error, either:

- Replace the synonym or literal with a constant expression of the correct type.
- Define a synonym or literal of the correct type.

(Z.100: 2.2.2)

WARNING 68 Optional parameter of sort %1 omitted

A missing parameter may lead to uninitialized values, make sure it is not left out by mistake.

You get this warning if:

- You omit an IN parameter in a procedure call.

To remove the warning, either:

- Supply a constant value or a variable to the parameter in the procedure call.
- Remove the parameter from the procedure and from all calls to the procedure.

WARNING 69 Implicit remote variable, consider making it explicit

ERROR 70 Invalid import expression

The import identifier in an import expression is erroneous.

You get this error if:

- You use an integer where a boolean is expected.

To remove the error, either:

- Use a boolean value instead.
- Make the expected value an integer.

(Z.100: 4.13)

ERROR 71 Signal %1 is shadowed in connection

You get this error if:

- You have multiple declarations of a signal (same name) and two channels or signal routes that are connected uses different versions of a carried signal.

To remove the error, either:

- remove the inner declaration of the signal
- Rename one of the signals and determine which one the channels or signal routes in the connection should use.

ERROR 72 Invalid variable or attribute of variable

Reference to a nonexisting variable or a variable with illegal type or attribute from a view definition, a view expression or an export action.

You get this error either if:

- You EXPORT a variable that is not visible from the current scope.
- You EXPORT a variable that is not defined as EXPORTED.

To remove the error, either:

- Find the correct name of the variable you want to export.
- Make sure that the variable you want to export is defined as EXPORTED.

ERROR 73 Illegal use of parameterized signal

This message is output when context parameters are used in an SDL structure. Context parameters are not supported.

ERROR 74 Illegal use of parameterized sort

This message is output when context parameters are used in an SDL structure. Context parameters are not supported.

ERROR 75 Undefined timer

Use of an undefined timer in set, reset or active.

You get this error if:

- You are referring to a timer that has not been declared.

To remove the error, either:

- Declare the timer.
- Refer to an already existing timer instead.

(Z.100: 2.8)

ERROR 76 No range condition equals true in transition option

Exactly one of the branches out from a transition option must be true.

You get this error if:

- There is no alternative in a transition option that evaluates to true.

To remove the error, either:

- Change the condition for taking one of the alternatives to make sure that one alternative is always taken.
- Add a new alternative with a condition that evaluates to true.

(Z.100: 4.3.4)

ERROR 77 Undefined sort

A sort which is not defined is referenced.

You get this error if:

- You use a sort that has not been defined.

To remove the error, either:

- Use another sort that is already defined.
- Define the sort.

(Z.100: 2.2.2)

ERROR 78 Error not allowed where constant required

You get this error if:

- You use SDL keyword error in an expression required to be constant.

To remove the error:

- Replace the SDL keyword error with something else.

(Z.100: 5.3.9.1, 5.4.2.1)

ERROR 79 Variable expected

A variable was expected at this place.

You get this error either if:

- You use a constant value or an operator on the left side of the assignment.
- You give a constant value or the value of an operator instead of a variable as an actual parameter in a procedure call, when the corresponding formal parameter is an IN/OUT parameter.

To remove the error in the above cases, either:

- Use a variable on the left side of the assignment instead.
- Use a variable as an actual parameter in the procedure call instead.

You also get this error either if:

- You call a procedure using a constant as an actual in/out parameter.
- You call a procedure without specifying a parameter that is an in/out parameter.

To remove the error in the above cases, either:

- Change the constant or the not specified parameter to a variable of the correct type instead.
- Change the procedure to accept an in parameter instead of an in/out parameter.

(Z.100: 2.4.6)

ERROR 80 In parameter expected

An in parameter was expected according to the definition.

You get this error if:

- You are using an IN/OUT parameter in an imported procedure signature that should match an IN parameter in a remote procedure signature.

To remove the error, either:

- Change the imported procedure signature to use an IN parameter instead.
- Change the remote procedure signature to use an IN/OUT parameter instead.

ERROR 81 Unexpected parameter

The parameter should not be there according to the definition.

You get this error either if:

- You supply a parameter when you create a process that is not defined in an FPAR statement in the process definition.
- You add an extra parameter or return value to an imported procedure signature that should match a remote procedure signature.

To remove the error, either:

- Remove the extra parameter or return value.
- Add the extra parameter or return value to the definition of the called or created entity.

ERROR 82 Start transition required in instantiated type %1

An instantiated type must contain a start transition.

You get this error if:

- You try to instantiate a process type or a service type that has no start transition.

To remove the error, either:

- Instantiate a process type or a service type that has a start transition instead.
- Add a start transition to the process type or the service type.

ERROR 83 Start transition required in called procedure

You get this error if:

- You call a procedure that does not have a start transition.

To remove the error:

- Add a start transition to the procedure definition.

(Z.100: 2.4.6)

ERROR 84 Procedure call not allowed where constant required

You get this error if:

- You call a procedure in a place where a constant is required, such as in a variable or synonym definition.

To remove the error, either:

- Assign the variable a constant value in the variable definition.
- Assign the variable the return value of a procedure call in the start transition instead.

(Z.100: 5.3.1.9.1)

ERROR 85 Parameterized procedure cannot be called

(Z.100: 6.1.2)

ERROR 86 PId Expression or Process Identifier expected

An PId expression or a process identifier was expected at this place.

You get this error if:

- You supply a second parameter to import, but it is not a pid expression or a process identifier.

To remove the error, either:

- Change the second parameter to a pid expression or a process identifier.
- Remove the second parameter if it is not needed.

ERROR 87 Pid expression in procedure call only allowed for remote procedures

It is only allowed to have PId expressions in a procedure call if the procedure is a remote procedure.

You get this error if:

- You are using the TO keyword in a call to a procedure that is not remote and imported.

To remove the error, either:

- Remove the TO construct from the procedure call, if it is a normal procedure residing in this process.
- Make the procedure a REMOTE procedure in the other process, and declare the procedure as IMPORTED in this process.

(Z.100: 2.7.3, 4.14)

ERROR 88 Undefined procedure

Attempt to call an undefined procedure.

You get this error if:

- You call a procedure that is not defined or not visible from the current scope.

To remove the error, either:

- Make an existing procedure with the correct name visible from the current scope.
- Create an appropriate procedure if it does not exist.

(Z.100: 2.2.2, 2.7.3)

WARNING 89 Trailing parameter of sort %1 omitted

A missing parameter may lead to uninitialized values, make sure it is not left out by mistake.

ERROR 90 Two range conditions equals true in transition option

The branches out from a transition option must be mutually exclusive.

You get this error if:

- You have two alternatives in a transition option that are both evaluated to be true.

To remove the error, either:

- Change the condition for taking one of the alternatives to make sure that only one alternative is taken.
- Remove one of the alternatives that evaluated to true.

(Z.100: 4.3.4)

INFO 91 Location of the global definition previously mentioned

ERROR 92 Undefined remote variable

Attempt to use an undefined remote variable.

You get this error if:

- You use a remote variable that has not been defined.

To remove the error, either:

- Use another remote variable that is already defined
- Define the remote variable.

(Z.100: 4.13)

ERROR 93 Label expected on first item in free action

(Z.100: 2.6.7)

ERROR 94 Formal parameters required in operator definition

The operator definition must have formal parameters.

You get this error if:

- You have defined an operator that is missing a FPAR statement.

To remove the error:

- Introduce an FPAR statement in the definition of the operator.

(Z.100: 5.3.2)

ERROR 95 Recursive definition of parent sort %1 in syntype

A syntype must not be based on itself, directly or indirectly.

You get this error if:

- You define a syntype X that equals a syntype Y that equals syntype X.

To remove the error:

- Make sure that syntypes not equals each other in a circular chain, and that one syntype equals a non-syn-type.

(Z.100: 5.3.1.9)

INFO 96 Setlocale failed

The Analyzer failed to set the character handling part of the locale. It will remain in the default C locale.

ERROR 97 Undefined signallist

ERROR 98 New literal %1 must only occur once in literal renaming

All literals renamed in an inheritance must be given unique names.

You get this error if:

- You are declaring a newtype that inherits from another type, and renames two or more old literals to the same new name.

To remove the error:

- Make sure to use unique names for all renamed literals.

(Z.100: 5.3.1.11)

ERROR 99 Old literal %1 must only occur once in literal renaming

A literal must only be renamed once when inheriting.

You get this error if:

- You are renaming literals in a newtype that inherits from a super newtype, and the old literal names occur more than once in the list of renamings.

To remove the error:

- Make sure that the same old literal name only occurs once on the right side of the assignment in the list of renamings.

(Z.100: 5.3.1.11)

ERROR 100 Illegal redeclaration of operator signature

It is not allowed to have several operators with the same signature. This error may also occur if you use the same name twice in a sort with implicit operators, for example “structure”.

ERROR 101 Unexpected remote variable

Remote variable may not be used in [priority] input.

You get this error if:

- You are using a remote variable in a way that is not allowed, for instance as if it was a normal signal in an input construct.

To remove the error, either:

- Use the remote variable in a proper way instead: to update the value of the remote variable, use an IMPORT construct instead of trying to receive the value as if it was a signal.
- Change the remote variable to another more proper type for the intended use.

ERROR 102 Expression(s) in number of instances evaluates to error

(Z.100: 2.4.4)

ERROR 103 More than one sub signal is visible

You get this error if:

- You have two or more REFINED signals with the same name that is visible from the current place.

To remove the error:

- Rename (or remove) one of the REFINED signals, to get unique names.

(Z.100: 3.3)

ERROR 104 Undefined signal, timer, signallist, remote procedure or remote variable

No visible definition with proper entity kind found.
(Z.100: 2.2.2)

ERROR 105 Remote procedure definition missing

A definition of the remote procedure is missing.

You get this error if:

- You are referring to a procedure with an IMPORTED construct, but the procedure does not exist or is not defined as REMOTE.

To remove the error:

- Make sure there is a REMOTE definition of the procedure you are referring to in the IMPORTED construct.

ERROR 106 Recursive definition of synonym

The value of a synonym must not be based on itself, directly or indirectly.

ERROR 107 Gate must be connected

You get this error if:

- You instantiate a type without connecting all it's gates.

To remove the error:

- Connect the gate.

(Z.100:6.1.4)

ERROR 108 Type mismatch for variable or formal parameter

ERROR 109 Signal refinement is not supported

ERROR 110 Only one varargs allowed

ERROR 111 Unexpected return value

The procedure has to be value returning to use return values.

ERROR 112 Return value expected

A value returning procedure must specify a return value.

You get this error if:

- You are returning from a procedure without specifying a return value, when the procedure is expected to return a value because of a RETURNS construct in the procedure heading.

To remove the error, either:

- Supply a return value in the RETURN construct.
- Remove the RETURNS construct from the procedure heading and remove any returned values from all RETURN constructs in the same procedure.

ERROR 113 Undefined state %1

A nextstate contains a name of a state which is not defined.
(Z.100: 2.6.7.2.1)

ERROR 114 Undefined label %1

Attempt to join an undefined label (connector).
(Z.100: 2.6.8.2.2)

ERROR 115 Open range does not match

A range condition does not match the actual context.
(Z.100: 5.3.1.9.1)

ERROR 116 Inherited operator %1 is not visible

Operators containing an exclamation cannot be inherited (and renamed).

You get this error if:

- You declare a newtype that inherits an operator that cannot be found in the current scope.

To remove the error in the above cases, either:

- Check that you are using the correct name for the operator.
- Declare the operator.

You also get this error if:

- You have a newtype that inherits from a super newtype, and you try to inherit an operator from the super newtype that does not exist.

To remove the error in the above case, either:

- Make sure that the name of the operator you wants to inherit is correct.
- Remove the name of the non-existing operator from the list of inherited operators.

(Z.100: 5.3.1.11)

ERROR 117 Old operator name %1 must only occur once in operator renaming

An operator must only be renamed once when inheriting.

You get this error if:

- You use the same name for the old operator name and the new operator name.

To remove the error:

- Use a new name instead of reusing the old name.

(Z.100: 5.3.1.11)

ERROR 118 New operator name %1 must only occur once in operator renaming

All operators renamed in an inheritance must be given unique names.

You get this error if:

- You use the same new operator name for several old operator names.

To remove the error:

- Make sure to use unique names for every new operator name.

(Z.100: 5.3.1.11)

ERROR 119 Inheritance for %1 is circular

You get this error if:

- You have created a circular chain of diagram inheritance.

To remove the error:

- Break the chain of circular inheritance by removing or changing the super diagram for one of the diagrams in the chain.

(Z.100: 6.3.1)

ERROR 120 Atleast constraint for %1 is circular

(Z.100: 6.3.2)

ERROR 121 Recursive sort inheritance

A sort must not inherit from itself, directly or indirectly.

You get this error if:

- You have created a circular chain of newtype inheritance.

To remove the error:

- Break the chain of circular inheritance by removing or changing the super newtype for one of the newtypes in the chain.

(Z.100: 5.3.1.11)

ERROR 122 Unexpected connect statement

A connect statement was not supposed to be found at this place.

INFO 123 %1 is one of the possible matches

ERROR 124 Signal expected in output

An output must contain at least one signal.

You get this error if:

- You are trying to send something from a process that is not a signal, such as a timer.

To remove the error, either:

- Change the name in the output construct to match a visible signal instead.
- Supply the value or variable that you were trying to send as a parameter to a signal.

(Z.100: 2.7.4)

ERROR 125 Undefined channel or signal route in output via

Identifiers used in a via clause of an output must be visible channels or signal routes.

You get this error if:

- You use a name of a non-existing channel or signal route after a VIA keyword in a signal sending.

To remove the error, either:

- Find a correct channel or signal route name and use that instead.
- Add the channel or signal route that you references in the VIA construct.
- Remove the VIA construct, if the VIA construct is not needed and the signal sending path can be determined uniquely anyway.

(Z.100: 2.7.4)

ERROR 126 Set with no time expression is only allowed for timers with default duration

It is only allowed to omit the time expression in set if there is a default duration specified for the timer.

You get this error if:

- You try to set a timer that does not have a default duration without specifying a duration in the SET construct.

To remove the error, either:

- Add a default duration to the timer definition:
  `TIMER myTimer:=10.`
- Specify a time in the SET construct: `SET(Now+10, myTimer).`

(Z.100: 2.8)

ERROR 127 Set must contain a time expression and a timer id

When using set you must specify when the timer should expire and which timer to set.
(Z.100: 2.8)

ERROR 128 Undefined process

Attempt to create an undefined process.

You get this error if:

- You are trying to create a process that has not been defined.

To remove the error, either:

- Check that you are using the correct name for the process that you want to create.
- Add a process definition of the process you want to create.
- Remove the process creation construct.

(Z.100: 2.2.2, 2.7.2)

ERROR 129 Created process must belong to the partition block %1

It is only allowed to create a dynamic instance of a process type if the process type definition is in the same block.

You get this error if:

- You try to create a process instance in another block than the block that the creating process resides in.

To remove the error:

- Make sure that the creating process and the process to be created resides in the same block diagram.
- Make sure that the creating process uses the correct name for the process to be created.

(Z.100: 2.7.2, FD)

ERROR 130 Remote procedure input and signal list input must not have parameters

It is not allowed to have parameters in a remote procedure input or when inputting a signal list.

You get this error if:

- You try to receive parameters by using a pair of parenthesis characters when referring to a remote procedure or a signal list in an INPUT construct.

To remove the error, either:

- Replace the remote procedure or the signal list with ordinary signals
- Remove the parameters and the associated pair of parenthesis characters.

(Z.100: 4.14)

WARNING 131 %1 : Signal not used in input or output

The defined signal has neither been used in an input nor in an output.

ERROR 132 Invalid procedure call

Expressions in Enabling condition and continuous signal must not contain procedure calls.

You get this error if:

- You are calling a procedure in an inappropriate place, such as in the expression related to a continuous signal.

To remove the error:

- Remove or move the procedure call from the inappropriate place.

ERROR 133 State name %1 in asterisk state list must be contained in other state lists

The state names used within the parenthesis (the exceptions) of an asterisk state must be defined somewhere in the enclosing body or the body of a supertype.

You get this error if:

- You include a name of a non-existing state in the list of exceptions after an asterisk (i.e. for all states) state.

To remove the error, either:

- Check that you are using the correct name for the state that should be excepted.
- Remove the unwanted exception.
- Introduce the missing state in your state machine.

(Z.100: 4.4)

ERROR 134 No state left to expand the asterisk with

The asterisk state list in the asterisk state includes all states in the enclosing process/procedure/service body.

You get this error if:

- You are using an asterisk (i.e. for all states) state that is representing zero states.

To remove the error, either:

- Make sure that you are not doing exceptions to the asterisk state for all states in the state machine.
- Replace the asterisk state with a normal state not using an asterisk.

(Z.100: 4.4)

ERROR 135 At least one state list must be different from asterisk

A process/procedure/service body cannot only contain asterisk states.

You get this error if:

- You only have asterisk states in the state machine.

To remove the error:

- Make sure that you have at least one non-asterisk state in the state machine.

(Z.100: 4.4)

ERROR 136 Several virtual continuous signals in a state with same or no priority

It is not possible to tell which transition would be affected by a redefinition.

You get this error if:

- You have several virtual continuous signals in the same state with the same priority.

To remove the error, either:

- Remove one of the virtual continuous signals.
- Give one of the virtual continuous signals another priority.

(Z.100: 6.3.3)

ERROR 137 Virtual continuous signal is virtual in super type with same or no priority

It is not possible to tell which transition would be affected by a redefinition.

You get this error if:

- You have a virtual continuous signal in both the sub type and the super type and the priority is the same or no priority is given.

To remove the error, either:

- Make the continuous signal in the sub type REDEFINED or FINALIZED instead of VIRTUAL.
- Give different priority to the continuous signal in the sub type than the priority in the super type.

(Z.100: 6.3.3)

ERROR 138 Redefined continuous signal is not virtual in super type with same or no priority

Cannot find the transition to redefine.

You get this error if:

- You redefine a continuous signal transition in a sub type and the same continuous signal transition cannot be found in a super type with the same priority.

To remove the error, either:

- If you intend to redefine an existing continuous signal, make sure the continuous signal you are redefining is starting from the same state and have the same priority.
- If you do not want to redefine an existing continuous signal, remove the REDEFINED or FINALIZED keyword.

(Z.100: 6.3.3)

ERROR 139 Filter exited with error code

A filter program terminated with a non zero exit code.

ERROR 140 External procedure not allowed

An external procedure cannot be mentioned in a <type expression>, in a <formal context parameter> or in an <atleast constraint>.

You get this error if:

- You try to use an external procedure where only a normal procedure is allowed.

To remove the error, either:

- Use a normal procedure instead of the external procedure.
- Make the external procedure a normal procedure.

ERROR 141 Number of block instances not allowed

Number of block instances may only be specified in typebased block definitions.
(Z.100: 6.1.3.3)

ERROR 142 No diagram to put definition in

It is not possible to convert something to a text symbol in SDL/GR without a surrounding diagram.

ERROR 143 Start is already defined in super type

It is not allowed to have more than one start transition and a start transition is already defined in the inherited type.

You get this error if:

- You redefine a transition from a super type without using the keyword REDEFINED or FINALIZED.

To correct the error, either:

- Use the keyword REDEFINED or FINALIZED for the redefined transition, if you intend to redefine the transition; make sure that the transition you redefine is either marked with VIRTUAL or REDEFINED.
- Remove the definition from the sub type if you do not intend to redefine it.

(Z.100: 2.6.2, 6.3)

ERROR 144 Start is defined as finalized in super type

It is not allowed to define the start transition as redefined, since it is defined as finalized in the inherited type.

You get this error if:

- You are trying to REDEFINE or FINALIZE the start transition in a sub type when it is declared as FINALIZED in a super type.

To remove the error, either:

- Change the start transition to VIRTUAL or REDEFINED in the closest super type where it is mentioned.
- Remove the start transition from the sub type if you did not intend to redefine the start transition.

(Z.100: 2.6.2, 6.3.3)

ERROR 145 Start is not defined as virtual in super

It is not allowed to define the start transition as redefined, since it is not defined as virtual in the inherited type.

You get this error if:

- You try to redefine the start transition in a sub type when the start transition is not defined as VIRTUAL in a super type.

To remove the error:

- Make the start transition VIRTUAL in the super type.

(Z.100: 2.6.2, 6.3.3)

ERROR 146 Multiple exits from state %2 with signal, timer or remote procedure %1

The signal, timer or remote procedure is contained in two inputs or one input and one save.

You get this error if:

- There are several ways to leave a state for the same signal, timer or remote procedure.

To remove the error:

- Remove enough ways to leave the state to make all remaining ways unique.

(Z.100: 2.6.3, 2.6.5)

ERROR 147 Virtual input %1 is already defined as input in state %2 in supertype

It is not allowed to define the input %1 as virtual, since it is already defined without virtual in the inherited type.

You get this error if:

- You are declaring a transition to be VIRTUAL in a sub type, when the transition is already VIRTUAL in the super type.

To remove the error:

- Use the keyword REDEFINED instead of VIRTUAL in the sub type.
- Make sure the transition is VIRTUAL in a super type.

(Z.100: 6.3.3)

ERROR 148 Redefined input %1 is not defined as virtual input in state %2 in supertype

It is not allowed to define the input %1 as redefined, since it is not defined with virtual in the inherited type.

You get this error if:

- You have a redefined transition in a sub type that does not exist in the super type.
- You have a redefined transition in a sub type that is not declared as virtual in the super type.

To remove the error, either:

- Make sure there is a matching virtual transition in the super type.
- Remove the REDEFINED keyword from the transition in the sub type.

(Z.100: 6.3.3)

ERROR 149 Redefined input %1 is defined as finalized input in state %2 in supertype

It is not allowed to define the input %1 as redefined, since it is defined as finalized in the inherited type.

You get this error if:

- You are trying to REDEFINE a transition that is marked as FINALIZED in a super type.

To remove the error, either:

- Change the transition from FINALIZED to REDEFINED in the super type.
- Make sure that you are redefining the correct transition.

(Z.100: 6.3.3)

ERROR 150 Redefined input %1 is defined as input in state %2 in supertype

It is not allowed to define the input %1 as redefined, since it is defined without virtual in the inherited type.

You get this error if:

- You define a transition as REDEFINED in a sub type, when the same transition in the super type is not VIRTUAL.

To remove the error, either:

- Make the transition in the super type VIRTUAL.
- Make sure that you are redefining the correct transition.

(Z.100: 6.3.3)

ERROR 151 Finalized input %1 is not defined as virtual input in state %2 in supertype

It is only allowed to define an input as finalized if it is a virtual defined input in a inherited type.

You get this error either if:

- You use the keyword REDEFINED or FINALIZED for a transition that is not declared VIRTUAL in a super type.
- There is no super type.

To remove the error, either:

- Make the transition in the super type VIRTUAL.
- Change the name in the sub type to match the correct transition in the super type
- Make sure the sub type inherits from the correct super type.

ERROR 152 Finalized input %1 is defined as finalized input in state %2 in supertype

It is not allowed to define the input %1 as finalized, since it is already defined as finalized in the inherited type.

You get this error if:

- You try to finalize a transition in a sub type that is already FINALIZED in a super type.

To remove the error, either:

- Change the keyword FINALIZED to REDEFINED in the super type.
- Make sure that you are trying to finalize the correct transition in the sub type.

(Z.100: 6.3.3)

ERROR 153 Signal %1 is already defined as input in state %2 in supertype

It is not allowed to have the same signal in more than one input to the same state. In this case there is already a input of the signal %1 in the type you inherit from.

You get this error if:

- You try to define a transition that already exists in a super type, without using the keyword REDEFINED or FINALIZED, or when the transition in the super type is not virtual.

To remove the error:

- Add the keyword REDEFINED or FINALIZED to the transition in the sub type.
- Make sure that the transition in the super type is VIRTUAL.

ERROR 154 A state (%1) cannot contain both asterisk input and asterisk save

It is not allowed to both have a asterisk input and a asterisk save to a state.

You get this error if:

- You have a state in your state machine that has both an asterisk input transition and an asterisk save.

To remove the error, either:

- Remove the asterisk save.
- remove the asterisk input transition.

(Z.100: 4.6, 4.7)

ERROR 155 A state (%1) may only contain one asterisk input

It is only allowed to have one asterisk input to a state.

You get this error if:

- You are having more than one asterisk input transition for the same state.

To remove the error:

- Remove all but one asterisk input transition for the same state.

(Z.100: 4.6)

ERROR 156 A state (%1) may only contain one asterisk save

It is only allowed to have one asterisk save to a state.

You get this error if:

- You have a state in your state machine that has more than one asterisk saves.

To remove the error:

- Reduce the number of asterisk saves for the transition to one (or zero).

(Z.100: 4.7)

ERROR 157 Valid input signal set must be specified (in process/service %1) when no signal routes or channels are specified in the enclosing block/process (%2)

If there are no signal routes or channels specified to the process/service. “Signalset” must be used inside the process/service extended heading to specify the valid input signal set.

You get this error if:

- You have a process or service that is not connected with the outer world via signal routes or channels, and you have not specified an input SIGNALSET.

To remove the error, either:

- Add signal routes and channels to connect the process to the outer world.
- Add a SIGNALSET construct to the process and indicate the signals the process can receive.

(Z.100: 2.5.2)

ERROR 158 Value returning procedure expected

A value returning procedure was expected at this place.

ERROR 159 Type mismatch: sort %2 does not match %1

You get this error if:

- You are using an integer where a boolean is expected. For instance, you are using an integer as a type for a parameter in an imported procedure signature that should match a boolean type parameter in a remote procedure signature.

To remove the error, either:

- Use a boolean instead.
- Change the expected type to be an integer instead.

ERROR 160 In/out parameter expected

An in/out parameter was expected according to the definition.

You get this error if:

- You are using an IN parameter in an imported procedure signature that should match an IN/OUT parameter in a remote procedure signature.

To remove the error, either:

- Change the imported procedure signature to use an IN/OUT parameter instead.
- Change the remote procedure signature to use an IN parameter instead.

ERROR 161 Missing parameter

A parameter is missing according to the definition.

You get this error if:

- You omit a parameter or return value from an imported procedure signature that should match a remote procedure signature.

To remove the error, either:

- Add the missing parameter or return value to the imported procedure signature.
- Remove the not wanted parameter or return value from the remote procedure signature.

ERROR 162 Virtual definitions only allowed in types

It is only allowed to use virtual in a type definition.

You get this error if:

- You have a virtual definition of a diagram in a normal diagram.

To remove the error:

- Remove the keyword virtual from the definition of the diagram.
- Make the diagram with the virtual definition a type-based diagram.

ERROR 163 Virtual definition %1 does not conform to its virtuality constraint

(Z.100: 6.3.2)

ERROR 164 Redefinition of %1 does not conform to the virtuality constraint

(Z.100: 6.3.2)

ERROR 165 A gate with addition of signals can only occur in subtypes

Adding in gate definition can only be used if the type is inherited from another type.

You get this error if:

- You are using the ADDING construct for a gate in a diagram that does not inherit from another diagram.

To remove the error, either:

- Remove the ADDING construct from the gate definition.
- Make the type-based diagram with the error inherit from another type-based diagram with an appropriate gate that you can use.

(Z.100: 6.1.4)

ERROR 166 Definition of %1 exists already in super type

An already defined gate cannot be defined again. If you want to add signals to an already existing gate, the word adding must be used.

You get this error if:

- You declare a gate in a type that inherits from another type that already has a gate with the same name.

To remove the error in the above case:

- Rename one of the gates to get two gates in the sub type.
- Use the keyword ADDING in the gate in the sub type if you want to redefine the gate in the super type from the sub type.

You also get this error if:

- You use the RETURNS construct in both the super procedure type and the sub procedure type.

To remove the error in the above case:

- Remove the RETURNS construct from the sub type.

You also get this error either if:

- You try to redefine a finalized procedure in a sub type.
- You try to define a procedure, that has already been defined as virtual in a super type, without using REDEFINED.

To remove the error in the above cases, either:

- Make sure that you are redefining the correct procedure.
- Make sure that you use the REDEFINED keyword in the sub type, if you intend to redefine a procedure.
- Change the name of the procedure in the sub type, if you do not intend to redefine any existing procedure.
- Change the finalized procedure in the super type to be a redefined procedure.

`ERROR 167 A gate with addition of signals must already be defined in a super type`

Adding in gate definition cannot be used when defining a new gate.

You get this error if:

- You declare a gate endpoint constraint with the keyword ADDING, but the gate is not already declared in any super type.

To remove the error, either:

- Make sure that the name of the gate with the ADDING keyword matches the name of a gate in a super type.
- Remove the ADDING keyword to get a new gate that does not depend on any gate in a super type.

(Z.100: 6.1.4)

ERROR 168 Two gate constraints may not use the same direction in gate %1

It is not allowed to have the same direction (in or out) in two gate constraints.

You get this error if:

- You have two gates in the same type-based diagram going in the same direction.

To remove the error, either:

- Reverse one of the gates.
- Merge the two gates, i.e. remove one of them, if they are intended to go in the same direction.

(Z.100: 6.1.4)

ERROR 169 Undefined type %1 in gate endpoint constraint

The endpoint constraint in the gate contains a undefined or not visible type.
(Z.100: 6.1.4)

ERROR 170 The two endpoint constraints of gate %1 is not consistent

The endpoint constraints, to and from, in the gate are not consistent.

You get this error if:

- You have two gate endpoint constraints in a type-based diagram that are going in the same direction.
- You have two gate endpoint constraints in a type-based diagram that are not referencing the same outer entity.

To remove the error:

- Make sure that the two gate endpoint constraints are going in different directions.
- Make sure that if you reference an outer type-based diagram, the same type-based diagram is referenced in both gate endpoint constraints.

(Z.100: 6.1.4)

ERROR 171 Undefined block, channel or signal route in connection

Identifiers used in a connection must be visible in the scope and of the above mentioned types.
(Z.100: 2.5.3, 3.2.3)

ERROR 172 Undefined block, process or service in path

The identifiers following the text “from” and “to” in channel and signal route definitions must be visible blocks, processes or services.
(Z.100: 2.5.1, 2.5.2)

ERROR 173 Selected diagram not in an SDL system

The context must be available when analyzing part of a system.

ERROR 174 Undefined type %1 in instance specification

The type must be defined and visible in the scope to be instantiated.

You get this error if:

- You refer to a type-based diagram that does not exist.

To remove the error, either:

- Refer to an existing type-based diagram instead.
- Create a matching type-based diagram.

ERROR 175 Undefined component %1/%2 in use clause

(Z.100: 2.4.1.2)

ERROR 176 Parameterized type %1 cannot be used as constraint type

It is not allowed to use a type with context parameters as a constraint type.
(Z.100: 6.3.2)

ERROR 177 Undefined type %1 in constraint specification

The type %1 used in the constraint specification is undefined or not visible.

ERROR 178 Substructure name in subtype and supertype must be the same

(Z.100: 6.3.1)

ERROR 179 Undefined type %1 in inheritance specification

A type which is inherited is not defined or not visible.

You get this error if:

- You in a signal declaration uses the keyword INHERITS and mention another signal that has not yet been defined.

To remove the error:

- Make sure the signal to inherit from is defined before the signal that inherits.

ERROR 180 Undefined type %1 in virtuality constraint

The type %1 used in the virtuality constraint is undefined or not visible.

You get this error if:

- You refer to a type-based diagram that does not exist in a virtuality constraint.

To remove the error, either:

- Use the name of an existing and matching type-based diagram instead.
- Create a matching type-based diagram.
- Remove the virtuality constraint.

ERROR 181 Environment must only appear in one connection

When a channel is partitioned and has the environment as one of its endpoints, the channel substructure must contain one and only one connection with environment.

You get this error if:

- You have more than one connect statement for a signal route or channel going to or from the environment.

To remove the error:

- Remove all but one connect statement going to or from the environment for the signal route or channel in question.

(Z.100: 3.2.3)

ERROR 182 The path %1 must only appear in one connection

It is not allowed to have the same channel or signalroute name mentioned in more than one connection.

You get this error if:

- You mention a signal route or a channel in two or more CONNECT statements in the same diagram.

To remove the error:

- Merge all CONNECT statements mentioning the same signal route or channel into one CONNECT statement.

(Z.100: 2.5.3, 3.2.2)

ERROR 183 All subsignals to the channel %1 must be included in the signal lists of the subchannels

In the connection between a channel and subchannels, each signal conveyed by the channel must either be conveyed by the subchannels or all of its subsignals must be conveyed by the subchannels.
(Z.100: 2.5.3, 3.3)

ERROR 184 Signal %1 must be included in the signal list of the inner paths

In a connection point, all signals in the outer paths must be conveyed by at least one of the inner paths.

You get this error if:

- You connect two signal routes or channels that do not allow the same set of signals to be sent in each direction.

To remove the error:

- Add or remove signals from the signal lists in the connected signal routes or channels to get matching signal lists.

(Z.100: 2.5.3, 3.3)

ERROR 185 Signal %1 must be included in the signal list of the outer paths %2

In a connection point, all signals in the inner paths must be conveyed by at least one of the outer paths.

You get this error if:

- You connect a signal route or channel to a signal route or channel one level up in the diagram hierarchy that does not include all signals of the signal route or channel in this diagram.

To remove the error, either:

- Add the signals you want to send to the appropriate signal lists in the diagrams outside this diagram.
- Remove the not wanted and troublesome signal from the signal list of the channel or signal route that is connected to the world outside this diagram.

(Z.100: 2.5.3, 3.3)

ERROR 186 In the implicit connection with %2 signal %1 is not included in any of the outgoing implicit signal routes

Each signal in the channels, directed out from the block, must be mentioned in at least one output in a process. See also *ERROR 187*.
(Z.100: 2.5.2, 2.5.3)

ERROR 187 In the implicit connection with %2 is signal %1 not included in any of the incoming implicit signal routes

When a block contains no signal routes, the signal route definitions and the connections are derived from the valid input signal sets and the outputs (of the processes in the block) and the channels connected to the block. (The same is valid for a service decomposition with no service signal routes.) Each signal in the channels, directed into the block, must be included in at least one valid input signal set.
(Z.100: 2.5.2, 2.5.3)

ERROR 188 Actual context parameter %1 does not conform to the fpar constraint

The actual context parameter is not correct according to the fpar constraint given for the formal context parameter.
(Z.100: 6.2)

ERROR 189 Actual context parameter %1 does not conform to the sort signature constraint

The actual context parameter is not correct according to the sort signature constraint given for the formal context parameter.
(Z.100: 6.2)

ERROR 190 Actual context parameter %1 does not conform to the atleast constraint

The actual context parameter is not correct according to the at least constraint given for the formal context parameter.
(Z.100: 6.2)

ERROR 191 Actual context parameter %1 does not conform to the process type constraint

The actual context parameter is not correct according to the process type constraint given for the formal context parameter.
(Z.100: 6.2)

ERROR 192 Actual context parameter %1 does not conform to the sort list constraint

The actual context parameter is not correct according to the sort list constraint given for the formal context parameter.
(Z.100: 6.2)

ERROR 193 Actual context parameter %1 does not conform to the signal set constraint

The actual context parameter is not correct according to the signal set constraint given for the formal context parameter.
(Z.100: 6.2)

ERROR 194 Block definition must contain process, block or substructure definition

A block definition must at least contain one process or a substructure.
(Z.100: 2.4.3)

ERROR 195 Block definition must contain process definition(s) when containing signal route definition(s)

A block definition cannot contain signal routes when no processes are defined.
(Z.100: 2.4.3)

ERROR 196 Block definition must contain process or block definition(s) when containing connection(s)

A block definition cannot contain connections when no processes are defined.
(Z.100: 2.4.3)

WARNING 197 Expansion of select definition is not yet implemented

The SDL concept select definition is not supported by the SDL Analyzer.

> **Caution!**
>
> A consequence of this is that the parts of your SDL system which are contained in the select definition will **not** be analyzed or be input to the SDL to C compilers.

ERROR 198 Substructure definition must contain at least one block definition

At least one block must be defined inside a substructure.

You get this error if:

- You have a substructure diagram that does not contain any block diagrams.

To remove the error, either:

- Add at least one block diagram with one or several processes to the substructure diagram.
- Instantiate a block type diagram in the substructure diagram.

(Z.100: 3.2.2)

ERROR 199 Gate only allowed in paths with instance of a type or env in a type

Gates are only allowed when connecting channels or signal routes to types.

You get this error if:

- You are using gates without using type-based diagrams.

To remove the error in the above case, either:

- Remove the reference to the gate.
- Make the current diagram type-based, if you want to have gates to connect to in the environment.
- Make referenced diagrams instances of type-based diagrams, if you want to have gates to connect to in reference symbols.

You also get this error if:

- You use the VIA keyword when declaring a signal route or a channel, that is not going to or coming from an instantiation of a type-based diagram, and that is not going to or coming from the environment in a type-based diagram.

To remove the error in the above case:

- Remove the VIA construct.

(Z.100: 6.1.4)

ERROR 200 Undefined gate

Use of an undefined gate in a channel or signal route definition.

You get this error if:

- You are referring to a gate that does not exist.

To remove the error, either:

- Remove the gate reference.
- Add the gate you are referencing.

ERROR 201 Connected to gate (%1) in wrong scope

(Z.100: 6.1.4)

ERROR 202 Connected instance %1 does not conform to the endpoint constraint of gate %2

The type of the instance %1 connected to gate %2 must be equal to or be a subtype of the endpoint constraint of gate %2.

ERROR 203 ENV does not conform to the endpoint constraint of gate %1

A gate with an endpoint constraint cannot be connected to the environment.

You get this error if:

- You have a gate endpoint constraint in the type you try to instantiate, and you connect the gate to an instance that is not of the same type (or a descendant of that type) as the one mentioned in the gate endpoint constraint.

To remove the error, either:

- Connect the gate to an instance of the same type (or a descendant of that type) as the one mentioned in the gate endpoint constraint.
- Remove or change the gate endpoint constraint.

ERROR 204 Signal %1 in path is not an incoming signal of %3 gate %2

The gate %2 of instance %3 must be defined to convey the incoming signal %1. Add the signal to the gate definition.

ERROR 205 Signal %1 in path is not an outgoing signal of %3 gate %2

The gate %2 of instance %3 must be defined to convey the outgoing signal %1. Add the signal to the gate definition.

ERROR 206 Via gate expected in env path with block type or process type

A channel or a signal route must be defined using “via <gate>” when connected to the environment in a block type or system type.

You get this error if:

- You declare a channel or a signal route in a type-based diagram going to or coming from the environment without using a VIA keyword to specify the gate to use.

To remove the error:

- Add a VIA keyword and mention the gate to use in the declaration of the signal route or channel.

ERROR 207 Via gate expected in path with typebased instance

Defining a channel or a signal route using “via <gate>” is only possible when the connected instance is type based and thus contain gates.

You get this error if:

- You declare a signal route or a channel going to or coming from an instantiation of a type-based diagram, without specifying the gate to connect to with a VIA keyword.

To correct the error:

- Add a VIA keyword and the name of the gate in the instantiated type-based diagram that you want to connect to.

ERROR 208 Endpoints of the path %1 must be different

A channel/signal route cannot connect a block/process/service with itself.

You get this error if:

- You specify the same endpoint for both ends of a signal route or channel.

To remove the error:

- Connect one of the ends of the signal route or channel to another endpoint.

(Z.100: 2.5.1, 2.5.2)

ERROR 209 Endpoint %1 of channel %2 must be a block or process

A channel must be connected to at least one block!

(Z.100: 2.5.1)

ERROR 210 Endpoint %1 of the path %2 must be locally defined

The endpoints of a channel/signal route must be defined in the same scope as the channel/signal route is defined.

You get this error if:

- You try to connect a signal route or channel to a diagram that is not visible from the current diagram.

To remove the error:

- Connect the signal route or channel to the environment or to a diagram one level below the current diagram. If necessary, create more signal routes or channels in other diagrams to reach the diagram you want to communicate with.

(Z.100: 2.5.1, 2.5.2)

ERROR 211 The second path must denote reverse direction of the first path in %1

In a bidirectional channel or signal route, the second path must be in the reverse direction of the first path.

You get this error if:

- The second FROM-TO-VIA-WITH construct is not going in exactly the reverse direction as compared to the first one.

To remove the error, either:

- Make sure that the same connection item is used after both FROM in the first construct and after TO in the second construct.
- Make sure that the same connection item is used after both TO in the first construct and after FROM in the second construct.

(Z.100: 2.5.1, 2.5.2)

ERROR 212 Endpoint %1 of signal route %2 must be a process or a service

A signal route must connect at least one process or service.
(Z.100: 2.5.2)

ERROR 213 Return not allowed here

You get this error if:

- You use the return construct in a process diagram.

To remove the error, either:

- Replace the return with a stop.
- Make the process diagram a procedure diagram.

(Z.100: 2.6.8.2.4)

ERROR 214 Process definition must contain either processbody or service decomposition

Syntax requirement.
(Z.100: 2.4.4)

ERROR 215 Process cannot contain timer definition when decomposed into services

(Z.100: 2.4.5)

ERROR 216 Revealed and exported variables only allowed in process and service

(Z.100: 2.4.6)

ERROR 217 Complete semantic analysis requires a system

(Z.100: 2.4.6)

ERROR 218 Procedure cannot contain stop

Syntax requirement.
(Z.100: 2.6.8.2.3)

ERROR 219 Undefined view

ERROR 220 Step expression requires a loop variable

It is not allowed to omit the variable indication from a for statement when having a step expression.

You get this error if:

- You have a for(x,y,z)-statement where x does not define a loop variable, but z uses a loop variable.

To remove the error, either:

- Define a loop variable in the for statement.
- Change the step expression to match the loop variable.

ERROR 221 Break without name or continue are only allowed within a loop

Break terminates a for loop and continue does the next iteration of a for loop. They can only be used in the context of for loops.

You get this error either if:

- You use a BREAK construct without a name outside a loop.
- You use a CONTINUE construct outside a loop.

To remove the error, either:

- Add a name after the break keyword that matches a label that you want to jump to.
- Remove the break or continue construct.
- Put the break or continue construct within a for loop.

ERROR 222 Variable used before it is defined

A variable may only be used in statements following its definition.

You get this error if:

- You use a variable before the declaration of the variable.

To remove the error, either:

- Move either the variable use or declaration to make sure that the variable declaration comes before the variable use.
- Make sure you are using the correct variable.

ERROR 223 A jump statement must be contained in a labeled statement with the given name

A jump statement is a more restrictive form of a join statement. The intent is to create less convoluted code.

You get this error if:

- You have a break construct to a label that is not visible from the break.

To remove the error:

- If the label and the break are on the same level, begin a pair of brackets after the label that contains the break.

ERROR 224 No receiver found

ERROR 225 Two services cannot have the same signal (%1) in their valid input signal set

The complete valid input signal sets of the service definitions within a process definition must be disjointed.
(Z.100: 2.4.5)

ERROR 226 Recursive package %1 in use clause

A package must not use itself.

You get this error if:

- There is a self-referential chain of uses of packages. For instance package A uses package B, and package B uses package A.

To remove the error:

- Break the self-referential chain of uses of packages by removing a USE statement in an appropriate place in the chain.

ERROR 227 Ending type keyword must match starting

A process type must be terminated with endprocess type and not just endprocess, and so on.

ERROR 228 Not Charstring sort and no proper operator (Length, ...)

ERROR 229 Service signal routes cannot be specified when no signal routes are specified in the enclosing block

(Z.100: 2.4.5)

ERROR 230 Service decomposition must contain at least one service definition

(Z.100: 2.4.4)

ERROR 231 No proper Length operator found

ERROR 232 Recursive include of file %1

The file is included recursively.

ERROR 233 Qualifier in component command not in SDL system

All parts (components) of a program must belong to the same system.

ERROR 234 More than one system

Can only analyze one system at a time.

ERROR 235 Code generation requires a system

Packages are not sufficient.

ERROR 236 Valid input signal set cannot contain a timer (%1)

You get this error if:

- You include a timer in the specification of the set of signals that a process can receive.

To remove the error:

- Remove the timer from the signal set construct.

(Z.100: 2.4.4, 2.4.5)

ERROR 237 Two signals (%1 and %2) in the complete valid input signal set are on different refinement levels of the same signal

(Z.100: 3.3)

ERROR 238 Two signals (%1 and %2) among the output-signals of the process are on different refinement levels of the same signal

(Z.100: 3.3)

ERROR 239 Signal %1 in save cannot be received

The signal is not included in the complete valid input signal set of the process/service.
(FD)

ERROR 240 Signal %1 in input cannot be received

The signal is not included in the complete valid input signal set of the process/service.

You get this error if:

- You try to receive a signal in an input, but you have not declared that anyone can send that signal to this process.

To remove the error, either:

- Check that you are trying to receive the correct signal.
- Make sure that the signal can be sent to this process via signal routes and channels, or by including the signal in a SIGNALSET construct for this process.

(FD)

ERROR 241 Error creating directory %1

A message from the operating system is following this and hopefully clarifies the problem.

ERROR 242 Signal %1 in priority input cannot be received

The signal is not included in the complete valid input signal set of the service.
(FD)

ERROR 243 Actual in/out param sort %1 does not match %2

The sort of actual and formal in/out parameters of procedures must be identical on the syntype level.

You get this error if:

- You call a procedure using a parameter that is not of the correct type.

To remove the error, either:

- Call the procedure with a parameter of the correct type.
- Change the procedure to accept a parameter of the wanted type.

ERROR 244 Call procedure from expression or supply variable parameter

A value returning procedure should be called from an expression.

You get this error if:

- You call a value-returning procedure from outside of an expression.

To remove the error, either:

- Call the value-returning procedure from an expression in a task instead.
- Call another, similar procedure that do not return a value.
- Force the procedure to not return a value.

WARNING 245 Trailing parameter of sort %1 omitted

ERROR 246 Last param of procedure called from expr is not in/out

Actual in/out procedure parameters may not be omitted.

You get this error if:

- You are calling a procedure in an expression, and the called procedure does not have a RETURNS statement.

To remove the error:

- Add a RETURNS statement to the procedure definition, and make sure the procedure returns a value of the correct type.

ERROR 247 Actual in/out param cannot be omitted

The actual parameter can be omitted if the formal is in.

You get this error if:

- You call a procedure without less actual parameters than all formal parameters declared for the procedure.

To correct the error, either:

- Add the missing actual parameters to match all formal parameters declared for the procedure.
- Remove any not wanted formal parameters from the declaration of the procedure.

WARNING 248 License %1 will expire in %2 days

This is a notification that some license will expire in the near future.

ERROR 249 Aborted by exit request

This indicate that the analyzer received a stop message

ERROR 250 Component and Thread commands requires a previous Program command

Issue a program command to set the name of the executable before adding threads or components to it.

ERROR 251 Change directory: %1: %2

An error occurred when changing default (working) directory.

ERROR 252 Get work dir: %1: %2

The analyzer failed to obtain current working directory.

ERROR 253 Connection with %1 is missing

Either a path is connected to a block/process but is not contained in a connection or a block is connected to a channel but is not contained in a connection in the channel substructure.
(Z.100: 2.5.3, 3.2.2, 3.2.3)

ERROR 254 Connection with environment is missing

When a channel is partitioned and has the environment as one of its endpoints, the channel substructure must contain a connection with the environment.

You get this error if:

- You omit a connect statement saying that a channel or signal route is connected to the environment, when the declaration of the channel or signal route indicates that it is connected to the environment.

To correct the error:

- Add a connect statement saying that the channel or signal route is connected to the environment.

(Z.100: 3.2.3)

ERROR 255 Transition must end with a terminator

If the terminator of a transition is omitted, then the last action in the transition must contain a terminating decision.
(Z.100: 2.6.4, 2.6.8.1)

ERROR 256 Post Master

A problem occurred when communicating with some other tool using the PostMaster.

WARNING 257 Statement not reached

SDL-92 allows dead code in transition. The warning is issued to notify the user of the dead code.

ERROR 258 Initial transition ends (directly or indirectly) with dash nextstate

A nextstate in the initial transition (of a process/ procedure/ service) must be specified with a state name.
(Z.100: 4.9)

ERROR 259 Ending name (%1) cannot be qualified (except for remote definitions)

You get this error if:

- You are using a qualifier when specifying the name of the diagram at the end of the diagram definition.

To remove the error:

- Remove the qualifier and keep just the name of the diagram.

(Z.100: 2.2.2)

ERROR 260 The outer channeldef must have the first connectionpoint (environment) as one of the endpoints

A channel substructure can only contain a connection with the environment if the partitioned channel is connected to the environment.
(Z.100: 3.2.3)

ERROR 261 Answers are not mutually exclusive

Exactly one answer in a decision must match the question. For instance, a boolean decision with more than 2 answers or 2 answers that are both true or both false will produce this message.

You get this error if:

- It is possible to take two or more paths after the decision for at least one given set of values.

To remove the error:

- Make sure that it is always only possible to take one path after the decision symbol.

ERROR 262 The outer channeldef must have the first connectionpoint (block) %1 as one of the endpoints

The block identifier in a connection in a channel substructure must identify one of the endpoints of the partitioned channel.
(Z.100: 3.2.3)

ERROR 263 One of the endpoints of path %1 must be the scope unit %2

A path that occurs as the first connection point in a connection, must be connected to the block/process that contains the connection.
(Z.100: 2.5.3, 3.2.2)

ERROR 264 The second connectionpoint %1 must be defined in the scope unit %2

A path occurring in the second part (i.e. the list of subpaths) in a connection must be locally defined.

You get this error if:

- You have a connection statement connecting an outer channel or signal route with a non-existing inner channel or signal route.

To remove the error:

- Change the name of the second connection point to match an existing inner signal route or channel.
- Add an inner signal route or channel with the correct name.

(Z.100: 2.5.3, 3.2.2)

ERROR 265 One of the endpoints of path %1 must be the environment

A path that occurs in the second part (i.e. the list of subpaths) of a connection must have the environment as one of its endpoints.

You get this error if:

- You use a signal route or a channel in a CONNECT statement, and the signal route or channel is not connected to the environment.

To remove the error, either:

- Check that the CONNECT statement is connecting the correct signal route or channel in this diagram.
- Change the signal route or channel in this diagram to go to the environment with one of its endpoints.
- Remove the connect statement.

(Z.100: 2.5.3, 3.2.2)

ERROR 266 System definition must contain at least one block or process definition

An SDL system must contain at least one block or process.

You get this error if:

- You have constructed an SDL system that does not contain any blocks or processes.

To remove the error, either:

- Add at least one block with one or several processes to your SDL system.
- Instantiate any existing block type diagrams.

(Z.100: 2.4.2)

WARNING 267 Optional parameter of sort %1 omitted

WARNING 268 Gate must be connected

ERROR 269 Expression evaluates to error

ERROR 270 Selected definition not allowed here

ERROR 271 All packages must either be before or after the system

The system should follow the package list according to SDL. For backwards compatibility the analyzer also accepts a package list after the system, but not both at the same time. In the Organizer this means that the icons for all packages used by a system should be above the system icon.

WARNING 272 No analysis is performed when the infile is empty

The SDL/PR input file cannot be empty.

WARNING 273 No semantic analysis is performed when the input is not a system or a package

Semantic analysis can only be performed on a system or a package.

ERROR 274 Compile and link exited with error code

The code generated by the SDL to C Compiler caused compile or link errors.

ERROR 275 Lost license

The connection to the license server is lost. You must wait until the Analyzer has regained access to the license server before resuming your work. If required, contact your system manager to get the problem solved.

ERROR 276 Cannot get license

No license was available when starting up Analyzer. You must wait until a license becomes available (which will occur when another user terminates his Analyzer).

ERROR 277 Cannot return license

The Analyzer could not return its license to the license pool. To return the license, you may need to stop and restart the license server.

INFO 278 Analyzer command could not be fully performed

The Analyzer did not perform all the passes that were ordered. The results from the Analyzer may not be what you expect. This message is output as a result from a syntactic or semantic error that in turn causes the remaining passes not to be executed.

ERROR 279 The number of block instances must be one or greater

The number of block instances in a multiple block instantiation must be greater than 0.

You get this error if:

- You instantiate a block type in a block reference symbol with a number of instances smaller than one.

To remove the error:

- Change the number of instances to one or more.

ERROR 280 Unreachable path %1 in VIA

The path %1 must be connected to the process instance.

You get this error if:

- You send a signal from a process and the static structure of the SDL system does not declare that you are allowed to send that signal the way you want.

To remove the error, either:

- Add signal routes and channels as needed (with your signal in the signal lists), to reach the wanted process or the environment.
- Remove the sending of that particular signal.

ERROR 281 Unreachable process instance set %1 in TO expr

There must be a path leading to the process instance set %1 from this process instance.

You get this error if:

- You send a signal from a process and the static structure of the SDL system does not declare that you are allowed to send that signal the way you want.

To remove the error, either:

- Add signal routes and channels as needed (with your signal in the signal lists), to reach the wanted process or the environment.
- Remove the sending of that particular signal.

WARNING 282 Several possible paths in VIA list

There are several paths that can convey the signal, one of them will be non deterministically chosen.

WARNING 283 Several possible paths

There are several paths that can convey the signal, one of the paths will be non deterministically chosen during execution.

WARNING 284 Set-Case-Sensitive should be the first command to the analyzer

ERROR 285 Hex string literal must have an even length

This applies to values for types that are strings of bytes.

ERROR 286 Bit string literal must have a length modulo 8

This applies to values for types that are strings of bytes.

ERROR 287 Literal must be one octet long

This applies to values for types that are one byte.

WARNING 288 No matching answer for %2

If a decision (or transition option) is executed with a question not covered by any answer it will be an error. (Z.100:2.7.5 and 4.3.4)

ERROR 289 Expression in transition option evaluates to error

One of the expressions in a transition option is erroneous.
(Z.100:4.3.4)

ERROR 290 External synonym not allowed

External synonyms are only allowed in a number of instances in process reference symbols and in heading symbols in process diagrams.

You get this error if:

- You use an external synonym in the condition for a transition option alternative.

To remove the error:

- Remove the external synonym from the condition for the transition option alternative in question.

ERROR 291 Range check failed

You get this error if:

- You specify a value not allowed for a syntype, like a negative value in a variable of sort Natural.

To remove the error, either:

- Change the value to one of the allowed ones.
- Change the syntype to allow the value.
- Use a different sort allowing the value.

(Z.100:5.3.1.9.1)

ERROR 292 Maximal number of instances equals zero

A process should have at least one instance.

You get this error if:

- You specify zero as the maximum number of instances.

To remove the error, either:

- Change the maximum number of instances to one or more.
- Remove the diagram with maximum number of instances set to zero.

(Z.100:2.4.4)

ERROR 293 Maximum number of instances is less than initial number

In the process instances definitions, the maximum number of instances must be greater or equal to the initial number.

You get this error if:

- You specify a maximum number of instances that is less than the initial number of instances.

To remove the error, either:

- Increase the maximum number of instances to at least the initial number of instances.
- Decrease the initial number of instances to less than or equal to the maximum number of instances.

(Z.100:2.4.4)

ERROR 294 Not allowed to change remote procedure in redefinition

(Z.100:2.4.6)

WARNING 295 Consider adding else answer

If a decision (or transition option) is executed with a question not covered by any answer it will be an error. (Z.100:2.7.5 and 4.3.4)

ERROR 296 Redefined or finalized procedure must be exported when base is

(Z.100:2.4.6)

INFO 297 Error limit reached, terminating

The number of diagnostics that the Analyzer has reported exceeds the value of the limit.

WARNING 298 Accumulated expression depth: %1 parts: %2 matches: %3

Semantic analysis of complex expression may cause long execution time. The execution time will increase in a fashion that is exponential rather than linear. If possible, try to reduce the complexity of expressions by breaking them down into multiple expressions.

WARNING 299 Expression depth: %1 parts: %2

See WARNING 298 Accumulated expression depth: %1 parts: %2 matches: %3 above.

ERROR 300 Command not found

The command that was input to the Analyzer's command interpreter was not recognized.

ERROR 301 Unexpected end of command: %1

The command that was input to the Analyzer does not contain all necessary parameters.

ERROR 302 Ambiguous command

The command that was input to the Analyzer's command interpreter was ambiguous. More characters need to be supplied to identify the command.

ERROR 303 Parameter expected

The command that was input to the Analyzer's command interpreter requires one or more parameters.

ERROR 304 Macro must not be part of a qualifier

A macro is not allowed to be part of a qualifier.

ERROR 305 Remote entity may only be exported once in a process

This applies to exported procedures and variables.

(Z.100:2.4.6 and 2.6.1.1)

ERROR 306 No input specified

There is no input set to the Analyzer using the set-input command.

WARNING 307 %1: unknown, line ignored!

The command line is ignored.

ERROR 308 More than one package match use statement

Use Modules to group systems together with their package lists to make package names unique within a module.

INFO 309 Location of the shadowed definition

ERROR 310 Selected item %1 is not connected to a file

The Organizer has ordered to analyze a diagram that is not connected to a file.

ERROR 311 Signal %1 found in opposite direction in connection

You get this error if:

- You connect two signal routes or channels where a signal is defined according to the signal lists to go in different directions on the inside and on the outside of the connection.

To remove the error:

- Reverse the direction of one of the signal routes or channels, at least for the signal that caused the error.

ERROR 312 Syntax error in rule %1, symbol %2 found but one of the following expected:

Syntax error!

ERROR 313 Syntax error, symbol %1 found but one of the following expected:

Syntax error!

ERROR 314 Lexical error in rule %1, symbol %2 found but one of the following expected:

Syntax error!

You get this error either if:

- You have typed a character or keyword that is not allowed in a CHARSTRINGLITERAL in a place where the syntax checker thinks there should be a CHARSTRINGLITERAL.
- You have used an SDL keyword as a name for a variable.

To remove the error in the above cases, either:

- Examine the pointed out character or keyword to see if it really belongs there.

- Try to find out why the syntax checker expects a CHARSTRING-LITERAL; did you intend to have a CHARSTRINGLITERAL there?.
- Rename any variable that you have given the same name as an SDL keyword.

You also get this error if:

- You have constructed a word or token out of single characters that does not match allowed word or tokens in SDL.

To remove the error in the above case:

- Check the word in question and replace any mistyped or misplaced characters.

ERROR 315 Lexical error, symbol %1 found but one of the following expected:

Syntax error!

ERROR 316 File %1 not found in environment variable %2 %3, check installation!

Check the installation. If the error persists, contact Telelogic Customer Support. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

ERROR 317 Outside connect item %1 is locally defined

You get this error if:

- Your connect statement mentions an inner channel or signal route instead of a channel or signal route outside the current diagram as the first connect item.

To correct the error, either:

- Swap places (if the second connect item is wrong too) on the inner and outer channel or signal route in the connect statement.
- If there is a name conflict with a locally defined item, then add a qualifier to the outer connect item.

WARNING 318 %1 : signal not used
The signal %1 should be used.

Here is a small explanation valid for all *<entity> not used* warning messages:

You get this error if:

- You declare an entity but do not use it.

To remove the error, either:

- Remove the declaration.
- Start using the entity.
- Rename the entity in the declaration to match a used entity of the same type that does not have a declaration.
- Change the type in the declaration to match a used entity with the same name but with a different type (that does not have a declaration).

WARNING 319 %1 : timer not used in input
The timer %1 should occur in an input.

WARNING 320 %1 : timer not used in set
The timer %1 should be set in the process.

WARNING 321 %1 : signallist not used
The signallist %1 should be used.

WARNING 322 %1 : variable not used
The variable %1 should be used.

WARNING 323 %1 : formal parameter not used
The formal parameter %1 should be used.

WARNING 324 %1 : synonym not used
The synonym %1 should be used.

WARNING 325 %1 : viewed variable not used
The viewed variable %1 should be used.

WARNING 326 %1 : remote variable not used
The remote variable %1 should be used.

WARNING 327 %1 : sort not used
The sort %1 should be used.

WARNING 328 %1 : generator not used
The generator %1 should be used.

WARNING 329 %1 : label not used
The label %1 should be used.

WARNING 330 %1 : system type not used
The system type %1 should be used.

WARNING 331 %1 : block type not used
The block type %1 should be used.

WARNING 332 %1 : process type not used
The process type %1 should be used.

WARNING 333 %1 : procedure not used
The procedure %1 should be used.

WARNING 334 %1 : package not used
The package %1 should be used.

ERROR 335
Reserved for future purpose.

ERROR 336 Input to Instance generator must be a system

ERROR 337 Operator diagram/definition not allowed where constant required

You get this error if:

- You call an operator in a place where a constant is required, for instance in a declaration of a variable.

To remove the error, either:

- Remove the operator call, and replace it with a constant if needed.
- Call a non operator definition operator.

WARNING 338 %1 : operator diagram/definition not used

WARNING 339 %1 : operator not used

WARNING 340 %1 : remote procedure not used

ERROR 341 No matching variable, formal parameter, synonym or literal

ERROR 342 Literal %1 in literal renaming is not defined in parent sort

ERROR 343 No matching field selection, array element access or prefix operator

ERROR 344 No matching quoted operator

You get this error if:

- You try to use a quoted operator with the wrong number of arguments or the wrong type of arguments.

To remove the error, either:

- Check that you are using two arguments for infix operators and one argument for monadic operators.
- Check that the operators you are using are of the correct type.

ERROR 345 No matching (. .)

ERROR 346 No visible variable or formal parameter with this name

ERROR 347 Type mismatch in component selection of variable or formal parameter

ERROR 348 No matching monadic operator

ERROR 349 No matching infix operator

INFO 350 %1 is one of the possible matching types

ERROR 351 No matching remote variable in variable definition

INFO 352 %1 is one of the visible definitions

ERROR 353

Reserved for future purpose.

ERROR 354

Reserved for future purpose.

ERROR 355 No matching remote variable in import definition

ERROR 356

Reserved for future purpose.

ERROR 357 No matching character string

ERROR 358 No matching choice primary

ERROR 359 Now not allowed where constant required

ERROR 360 No matching now expression

ERROR 361 Import expression not allowed where constant required

You get this error if:

- You use an import expression where a constant is required.

To remove the error:

- Replace the import expression with a constant.

ERROR 362 PId expression not allowed where constant required

You get this error if:

- You are using a pid expression where a constant is expected.

To remove the error:

- Replace the pid expression with a constant expression.

ERROR 363 No matching PId expression

You get this error if:

- You use a PId expression where a boolean value is expected.

To remove the error, either:

- Replace the PId expression with a boolean expression.
- Make the expected value to be of type PId.

ERROR 364 View expression not allowed where constant required

You get this error if:

- You are using a view expression where a constant expression is expected.

To remove the error:

- Replace the view expression with a constant expression.

ERROR 365 Timer active expression not allowed where constant required

ERROR 366 Any expression not allowed where constant required

ERROR 367 No matching dereferencing operator

ERROR 368 Data base error

An internal error occurred in the Analyzer. Please send a report to Telelogic Customer Support. Contact information for Telelogic Customer Support can be found in "How to Contact Customer Support" on page *iv in the Release Guide*.

ERROR 369 No matching field selection

You get this error if:

- You try to access a field in a struct using a name that does not represent any field in the struct.
- You try to access a field in a struct whose sort does not match the one expected.

To remove the error:

- Make sure you are using the correct name to access a field in the struct.
- Add a field in the struct with the wanted name.
- If required, reference a conversion operator to have the field match the sort required.

ERROR 370 Type mismatch for procedure return value

ERROR 371 Type mismatch for variable, formal parameter, synonym or literal

You get this error if:

- You assign a variable a value of the wrong type.

To remove the error:

- Assign the value to a variable of the correct type instead.

ERROR 372 No visible variable, formal parameter, synonym, literal or operator with this name

You get this error if:

- You use the name of a variable, formal parameter, synonym, literal or operator that is not defined in the current scope.

To remove the error, either:

- Change the name to match a defined entity.
- Define the entity.

ERROR 373 No visible synonym or literal with this name

You get this error if:

- You refer to a synonym or a literal that cannot be seen from the current scope.

To remove the error, either:

- Correct the spelling mistake, if it was not meant to be a synonym or literal.
- Add a synonym or literal with the given name.

ERROR 374 Variable or formal parameter not allowed where constant required

You get this error if:

- You are using a variable or formal parameter in a place where a constant expression is expected.

To remove the error:

- Replace the variable or formal parameter with a constant expression.

ERROR 375 Type mismatch for synonym or literal

You get this error if:

- You use a boolean synonym or literal where an integer synonym or literal is expected.

To remove the error:

- Replace the boolean with an integer.

ERROR 376 Only one reference to each diagram allowed

Several diagrams connected to the same file.

ERROR 377 Keyword call required for procedure call

You get this error if:

- You are calling a value returning procedure without using the CALL keyword.

To remove the error:

- Add the CALL keyword before the procedure name in the expression.

ERROR 378 Several macrodefinitions have the same name %1

The name of a macro must be unique.

WARNING 379 Macrodefinition %1 was never called

The macro %1 should be used.

ERROR 380 Names separated by %, %1%%2, where neither is a formal parameter nor MACROID in macrodefinition %3

(Z.100: 4.2.1)

WARNING 381 Formal parameter %1 is never used in macrodefinition %2

The formal parameter %1 should be used in the macro definition

ERROR 382 The macrodefinition name is %1, but the endmacro name is %2

The names in the macro definition and the endmacro must be equal.

ERROR 383 Two formal parameters with the same name %1 in macrodefinition %2

Two formal parameters of a macro cannot have the same name.

ERROR 384 No matching macrodefinition found

The corresponding macro definition is missing.

ERROR 385 A macro may not call itself

A macro definition cannot be circular.

WARNING 386 No macro found

The system does not contain any macros or calls.

ERROR 387 Start transition expected

A process or procedure definition must begin with a start transition.

You get this error if:

- You define a process without defining a start transition.

To remove the error:

- Add a start transition to the process.

ERROR 388 Illegal redeclaration of operator definition

You get this error if:

- You declare an operator twice.

To remove the error, either:

- Rename one of the operators if they are different and you need them both.
- Remove one of the operator definitions.

(Z.100: 5.3.2)

ERROR 389 No matching operator signature found

The operator must be defined for this signature.

ERROR 390 This unexpected

The keyword this is only allowed in a procedure call, process create or “output to”.

You get this error if:

- You use the keyword THIS in normal diagrams.

To remove the error, either

- Remove or replace the keyword THIS.
- Convert the diagram to a type-based diagram.

ERROR 391 Context parameter %1 not allowed

(Z.100: 6.2)

ERROR 392 Call this invalidated by adding parameters to sub type

Call this cannot be used when formal parameters are added in the sub-type.

You get this error if:

- You use the keyword THIS in a procedure call and the called procedure is a super type to a sub type that adds parameters to be used in a procedure call.

To remove the error, either:

- Make both the sub type and the super type use the same set of parameters by declaring all parameters in the super type if you intend to call the sub type.
- Remove the THIS keyword if you intend not to call the sub type.

INFO 393 Instance path to the problem in previous message

This message provides additional information that may be used to locate the source of error in the previous message. This message may appear repeated times, depending on the depth of the SDL structure where the previous error is detected.

ERROR 394 Package Predefined expected, check installation

File *predef.sdl* does not contain the package Predefined. The error message appears only when you have altered the Telelogic Tau installation in an improper way.

ERROR 395 Type mismatch for view expression

You get this error if:

- You view a variable of the wrong type. For instance, you assign a variable of type boolean the value of a viewed variable of type integer.

To remove the error, either:

- Change the type either of the assigned variable or the viewed variable to match each other.
- View another variable instead, that has the correct type.
- Assign to another variable instead, that has the correct type.

ERROR 396 Viewed variable %1 is not revealed in block %2

The viewed variable must be declared as revealed in block %2.

ERROR 397 Context parameters are not supported

Context parameters are not implemented. Consider using an alternative approach.

WARNING 398 One line external formalism name expected on same line as alternative keyword

Syntax error! When using alternative the name of the alternative data formalism must occur on the same line as the alternative keyword.

ERROR 399 Type mismatch for timer active expression

You get this error if:

- You are using the result of the ACTIVE call as if it had another type than boolean.

To remove the error:

- Make sure you are using the result of the ACTIVE call as a boolean value.

ERROR 400 Type mismatch for any expression