Rational. SDL and TTCN Suite

Release Guide

Release Guide

IBM Rational SDL and TTCN Suite 6.3

Release Guide

How to Contact Customer Support

Contacting IBM Rational Software Support

If the self-help resources have not provided a resolution to your problem, you can contact IBM® Rational® Software Support for assistance in resolving product issues.

> **Note:**
>
> If you are a heritage Telelogic customer, you can go to http://support.telelogic.com/toolbar and download the IBM Rational Telelogic Software Support browser toolbar. This toolbar helps simplify the transition to the IBM Rational Telelogic product online resources. Also, a single reference site for all IBM Rational Telelogic support resources is located at http://www.ibm.com/software/rational/support/telelogic/

Prerequisites

To submit your problem to IBM Rational Software Support, you must have an active Passport Advantage® software maintenance agreement. Passport Advantage is the IBM comprehensive software licensing and software maintenance (product upgrades and technical support) offering. You can enroll online in Passport Advantage from http://www.ibm.com/software/lotus/passportadvantage/howtoenroll.html

- To learn more about Passport Advantage, visit the Passport Advantage FAQs at http://www.ibm.com/software/lotus/passportadvantage/brochures_faqs_quickguides.html.
- For further assistance, contact your IBM representative.

To submit your problem online (from the IBM Web site) to IBM Rational Software Support, you must additionally:

- Be a registered user on the IBM Rational Software Support Web site. For details about registering, go to http://www-01.ibm.com/software/support/.
- Be listed as an authorized caller in the service request tool.

Submitting problems

To submit your problem to IBM Rational Software Support:

1. Determine the business impact of your problem. When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting.

   Use the following table to determine the severity level.

Severity	Description
1	The problem has a ***critical*** business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
2	This problem has a ***significant*** business impact: The program is usable, but it is severely limited.
3	The problem has ***some*** business impact: The program is usable, but less significant features (not critical to operations) are unavailable.
4	The problem has ***minimal*** business impact: The problem causes little impact on operations or a reasonable circumvention to the problem was implemented.

2. Describe your problem and gather background information, When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Rational Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?

   To determine the exact product name and version, use the option applicable to you:

   - Start the IBM Installation Manager and select **File > View Installed Packages**. Expand a package group and select a package to see the package name and version number.

   - Start your product, and click **Help > About** to see the offering name and version number.

- What is your operating system and version number (including any service packs or patches)?

- Do you have logs, traces, and messages that are related to the problem symptoms?

- Can you recreate the problem? If so, what steps do you perform to recreate the problem?

- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, or other system components?

- Are you currently using a workaround for the problem? If so, be prepared to describe the workaround when you report the problem.

3. Submit your problem to IBM Rational Software Support. You can submit your problem to IBM Rational Software Support in the following ways:

   - **Online**: Go to the IBM Rational Software Support Web site at https://www.ibm.com/software/rational/support/ and in the Rational support task navigator, click Open Service Request. Select the electronic problem reporting tool, and open a Problem Management Record (PMR), describing the problem accurately in your own words.

     For more information about opening a service request, go to http://www.ibm.com/software/support/help.html

     You can also open an online service request using the IBM Support Assistant. For more information, go to http://www-01.ibm.com/software/support/isa/faq.html.

   - **By phone**: For the phone number to call in your country or region, go to the IBM directory of worldwide contacts at http://www.ibm.com/planetwide/ and click the name of your country or geographic region.

- **Through your IBM Representative**: If you cannot access IBM Rational Software Support online or by phone, contact your IBM Representative. If necessary, your IBM Representative can open a service request for you. You can find complete contact information for each country at http://www.ibm.com/planetwide/.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Rational Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Rational Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Rational Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Documentation

The complete user documentation for the SDL Suite and TTCN Suite products is available on-line. Selected parts are delivered in printed format.

On-Line Documentation

The on-line documentation consists of HTML files, which are included on the CD-ROM distribution. On Windows platforms the on-line help is a compiled HTML help file.

The HTML files can be accessed from within SDL Suite and TTCN Suite as a result of a context-sensitive help request.

Printed Documentation

The online documentation is available in PDF format on the SDL Suite and TTCN Suite CD. Use them for quick viewing or for printing selected parts of the documentation.

Contents of the Included Volumes

The following volumes are included in the documentation:

- Installation Guide:

  Describes the installation of SDL Suite and TTCN Suite, how to configure the computer environment, and how to handle licensing.

- Release Guide (this volume):

  General information about the product release, new features, limitations, and compatibility issues. Compatibility is described between SDL Suite 6.3 and TTCN Suite 6.3 and previous versions, and also between supported languages and published standards of these languages.

- Methodology Guidelines:

  Practical guidelines on how to use SDL and related notations in software development using the SDL Suite. The SDL-92 object oriented concepts are described as well as handling of data types.

- Methodology Guidelines:

Describes a methodology that will assist you in taking advantage of the TTCN language in a TTCN Suite environment. The TTCN language, concepts and data types are described.

- Getting Started:

  Introduction to the SDL Suite, and the supported languages and notations. By reading and practising the included tutorials, you will get more familiar with the basic SDL Suite tools.

  This volume will also assist you in getting familiar with the SOMT method and Cmicro targeting. Lectures are combined with practical exercises.

- Getting Started:

  Introduction to TTCN and the TTCN Suite, as well as tutorials on the TTCN Suite tools that will help you in getting familiar with the TTCN Suite.

- User´s Manual:

  Reference information and guideline material about all tools and features. All windows, dialogs, menus, commands, files, storage formats, options, etc. are described. There are also guides on how to use the tools and perform various tasks.

Typographic Conventions

This section describes the typographic conventions used in all volumes of the user documentation.

Menu Commands and UI Objects

All user interface objects, such as labels, menu names, menu items and button texts, are shown in *italics*:

The *Edit* menu provides a *Paste* command...
Click the *OK* button...

Menu commands can also be expressed in a single form by separating the menu, submenu and menu item by a '>' character:

Select *Edit* > *Paste* to...
The command *Tools* > *Link* > *Traverse* is used for...

File Names, Variables, etc.

Names of files and directories, variables, source code literals, file keywords, etc. are usually shown in a `typewriter font`:

The `sdt.ssl` file...
Set the `$telelogic` variable...

Textual Input/Output, Listings, etc.

Listings of source code, file contents and textual output from the computer is shown in a typewriter font:

```
State1:
  Sig1:INPUT
  Sig2:SAVED
State2:
  Sig3:INPUT
  Sig1:UNHANDLED
```

When you are asked to type in characters from the keyboard during an interactive session, a bold typewriter font is used:

Now enter **`Convert, Demon, System`**

```
C:\IBM\Rational\SDL_TTCN_Suite6.3> mkdir example
```

Placeholders

Information within angle brackets "< >" denotes symbolic names (placeholders) which should be replaced by actual names:

> The syntax is "<*process instance*> <*variable name*>"...

Keys

Special keys or key combinations are shown in a typewriter font within angle brackets. Keys to be pressed together are separated by '+':

> To execute the command, press `<Return>`...
> Press `<Ctrl+C>` to...
> The shortcut is `<Ctrl+Meta+V>`...

Platform Differences

The documentation is common to UNIX and Windows. The differences that exist between the platforms are usually indicated in the text by a *platform indicator* (a bold, sans-serif font) which always contains the word **UNIX** or **Windows**. Examples:

> The information is stored in the file `.sdtpref` **(on UNIX)** or `sdt.ini` **(in Windows)**.
>
> XYZ consists of the following:
>
> - Alpha
> - Beta **(UNIX only)**
> - Gamma
> - Delta **(Windows only)**
> - Etc.

If a complete section of text is platform specific, the platform is indicated in the section heading:

Performing the Operation (on UNIX)

Performing the Operation (in Windows)

Screenshots are shown in either a UNIX or Windows version. Even if the appearance and layout may differ slightly between the platforms or different screen setups, the functionality is identical.

Legal Information

This edition applies to IBM Rational SDL Suite 6.3 and IBM Rational TTCN Suite 6.3 and to all subsequent releases and modifications until otherwise indicated in new editions.

Copyright Notice

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send written license inquiries to the following:

```
IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
```

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send written inquiries to:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
Intellectual Property Dept. for Rational Software
IBM Corporation
1 Rogers Street
Cambridge, Massachusetts 02142
U.S.A.
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.
If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Additional legal notices are described in the legal_information.html file that is included in your software installation.

Copyright License

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

Trademarks

See http://www.ibm.com/legal/copytrade.html.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.html.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Chapter 1 Compatibility Notes

In this chapter you can read about compatibility issues between earlier SDL Suite and TTCN Suite releases and the SDL Suite 6.3 and the TTCN Suite 6.3 releases, as well as between the 2.3 release and the 3.X releases. It also describes the compatibility issues for the supported languages in the SDL Suite.

This chapter concentrates on issues of storage formats and language support. For the TTCN Suite, this chapter describes how to upgrade an old TTCN Suite database, and how to open a TTCN Suite database with an old version of the TTCN Suite.

This chapter also describes known limitations and known issues for the SDL Suite and TTCN Suite.

Language Compatibility

Compatibility with ITU SDL

The SDL language support in the SDL Suite includes nearly all of SDL'92, as defined in the ITU-T Z.100 recommendation.

The main divergencies between SDL supported in the SDL Suite and the Z.100 recommendation are:

- The concept of a generic system is only partially supported.
- Macro calls are only allowed within flowchart diagrams.
- The names of diagrams and diagram pages must conform to the lexical rules of SDL, except that spaces are not allowed.

More detailed limitations can be found in "SDL Restrictions" on page 7 and "SDL Editor" on page 18.

The SDL Editor supports a few extensions to the SDL/GR notation:

- An *additional heading* symbol, intended for defining inheritance, specialization, formal parameters, signalsets, etc.
- An *operator reference* symbol, making operators visible in the SDL structure handled by the Organizer.

The SDL Suite supports several new SDL concepts, some of which are defined in an addendum to the latest Z.100 Recommendation, known as *SDL-96*, and some others which are IBM Rational-specific extensions to SDL. The extensions are supported in all relevant tools.

SDL-96 Extensions

- **Remote procedures on channels and signal routes:** This extension allows names of remote procedures to be put in signal lists of channels and signal routes. This has the main advantage that it enables processes to call remote procedures implemented in the environment of the SDL system, and also the other way around; remote procedures can be called from the environment of the SDL system.
- **External procedures:** External procedures are called like normal SDL procedures, but they are implemented in the target language (C). This makes it easier to use C functions in SDL.

- **External operators:** External operators are called like normal SDL operators, but they are implemented in the target language (C). This makes it easier to use C functions in SDL.

Other SDL-96 features are not supported in the SDL Suite.

SDL Suite specific Extensions

The following extensions to SDL are SDL Suite-specific. A detailed description of these extensions can be found in chapter 2, *Data Types* and chapter 3, *Using SDL Extensions, in the Methodology Guidelines*.

- **Choice:** With this construct you can define types with values where only one of a number of components can be present. Choice types replace the `#UNION` code generator directive (although `#UNION` is still supported for backward compatibility reasons).
- **Optional/default components in structs:** It is now possible to specify that fields of struct types are optional, or have a default value. This feature is useful in the design and implementation of protocols.
- **Size constraints:** With this construct it is possible to define types that should have a limited length.
- **Bit fields:** Integer fields in struct types can be specified as bit fields, very similar to C. This is useful when reusing C data types that contains bit fields in SDL.
- **Z.105 types and generators:** The following new predefined data types and generators from Z.105 have been added: Bit, Bit_String, Octet, Octet_String, Object_Identifier, IA5String, NumericString, PrintableString, VisibleString, Null, and the Bag generator.
- **Powerset operators:** Powerset has been extended with Length and Take operators, which make it possible to iterate over powersets and thereby make powersets more useful.
- **Pointers:** There are syntactic extensions that make it easier to work with pointers in SDL. In order to use pointers, the Ref generator from the package `ctypes` should be used.
- **Own and ORef:** The generators Own and ORef support safe pointers, making it possible to speed up applications by sending pointer

values instead of copying data during signal sending. Allocation and deallocation of memory is handled automatically.

- **Algorithms in SDL:** A number of new constructs have been added to allow algorithms to be expressed within a Task symbol, e.g. if-then-else, loops and decisions.
- **Extended operator:** Added support for operators without parameter, operators without result, and in/out parameters to operators (both operator signatures and diagrams). As an alternative to an assignment statement an operator application statement may invoke a non value returning operator.
- External variables supported.
- **Latin One:** ISO-Latin-1 supported in predefined sort character.
- **I18n:** Predefined sort charstring supports multi-byte character literals.
- **Names:** Underscore may be used without restrictions in names.
- A make operator has been added to the predefined items String, Powerset, Bag, and Object_identifier. The operator takes a variable number of arguments (0 and up). It is invoked by the “(. .)” notation. The signatures for the operators are:

```
make! : * Itemsort -> String;
make! : * Itemsort -> Powerset;
make! : * Itemsort -> Bag;
make! : * Natural -> Object_identifier;
```

  The * above indicate a variable argument (0 or more occurrences allowed).

- New operator in predefined type Object_identifier:

```
append    : in/out Object_identifier, Natural;
```

SDL-2000 Extensions

- Binary and hexadecimal literals for Integer, Bit_string, Octet, and Octet_string sorts. Examples ‘’b ‘1001’B ‘’H ‘Fe’h.

- Mix of block and process:
  - System may contain process (without intervening block).
  - Block may contain block (without intervening substructure).
  - Channels may be connected to processes.

Compatibility with ITU MSC

The MSC language support in the SDL Suite includes full support for *basic MSCs* (MSC'93), and also parts of MSC'96, as defined in the ITU-T Z.120 recommendation.

The key points for the MSC'96 support are:

- Roadmaps, or High-level Message Sequence Charts (HMSC), are supported in the HMSC Editor with the exception of the comment symbol and the parallel frame symbol.
- MSC diagrams support the reference symbol with MSC reference expressions, allowing references to other (H)MSC diagrams.
- MSC diagrams support qualifiers.
- MSC diagrams support inline expressions, allowing several scenarios to be expressed in the same diagram. Since the MSC Editor does not support gates, the inline expressions always cover all instances.
- Separate timer symbols are supported in MSC diagrams.
- MSC diagrams support lost and found messages (black and white holes).

Limitations to the MSC support can be found in "UML and MSC Restrictions" on page 9 and "MSC Editor" on page 17.

The MSC Editor also supports a few extensions to Z.120:

- Messages and timers may be assigned additional statuses that map the Z.100 recommendation, allowing to express a graphical trace in a more intuitive way.
- Messages are allowed to be *sent to self,* where the out event and the in event resides on the same instance (Z.120 does neither explicitly allow or forbid this).

MSC’96 Reserved Words

The following words are reserved in MSC’96:

action, all, alt, as, before, begin, block, by, comment, concurrent, condition, connect, create, decomposed, empty, end, endconcurrent, endmsc, endexpr, env, exc, expr, external, found, from, gate, in, inf, inline, inst, instance, loop, lost, msc, mscdocument, msg, opt, order, par, process, reference, related, reset, service, seq, set, shared, stop, subst, system, text, timeout, tim, to, via

The reserved words may not be used as names/identifiers even if the MSC Editor does not support the corresponding language construct; this to avoid compatibility problems either when the editors support more of the MSC’96 language or if the MSC/PR is to be exported to other tools with a different subset of the language supported.

MSC’93 Reserved Words

The following words are also treated as reserved words in the (H)MSC Editors but are mapped to (corresponding) MSC’96 reserved words.

MCS’93	MSC’96
instancehead	**instance**

This mapping is done to give old scripts and MSC/PR generators a chance to work.

Unsupported Reserved words

The following reserved words are not supported by the (H)MSC Editors (i.e. they are treated as reserved words with respect to names/identifiers but the Editors have no support for the constructs that can be created with these words).

before, begin, external, found, gate, lost, order, related, via

In addition, the SDL Explorer does not support the following reserved words:

subst, by, msg

Parameter Lists

Note that MSC/PR import allows empty parameter lists and an empty duration name when setting a timer, due to backward compatibility. This is not in accordance with either MSC’93 or MSC’96.

SDL Restrictions

This section describes the major SDL restrictions with respect to the ITU-T Z.100 recommendation of 1992. The restrictions lie in one or several specific tools. This means that even if a concept is not fully supported, it may sometimes be possible to use it in a limited part of the tool set.

This listing deals with restrictions in the following tools:

- SDL Editor
- GR to PR Converter
- SDL Analyzer
- SDL to C Compiler (Cbasic and Cadvanced)

For more detailed information on the SDL restrictions in the Analyzer and the SDL to C Compiler, see “SDL Analyzer” on page 20 and “Cadvanced SDL to C Compiler” on page 24.

For a description of SDL restrictions regarding the Simulator, the SDL Explorer, the TTCN Link kernels and the Cmicro SDL to C Compiler, please see the following sections in the User’s Manual:

- *“Restrictions” on page 2234 in chapter 49, The SDL Simulator*.
- *“Restrictions” on page 2388 in chapter 52, The SDL Explorer*.
- “SDL Restrictions” on page 1421 in chapter 35, *TTCN Test Suite Generation*.
- “SDL Restrictions” on page 3452 in chapter 65, *The Cmicro SDL to C Compiler*.

Z.100 Chapter	Concept and Restriction	Affected Tools
2.2.2 Visibility rules, names and identifiers	<name>: spaces in <name>s are not supported (a space is a shorthand for <underline>)	SDL Editor, Analyzer, SDL to C Compiler

Z.100 Chapter	Concept and Restriction	Affected Tools
2.4.1.2 Package	<interface> definitions, and <definition selection list> in use statements, are not supported.	Analyzer, SDL to C Compiler
2.5.1 Channel	**nodelay:** channels with delay and channels with nodelay are treated the same.	SDL to C Compiler
2.7.4 Output	**via all** to a <block instance set> only sends a single signal.	SDL to C Compiler
2.9 Internal input and output	The <internal input symbol> and the <internal output symbol> are not available.	SDL Editor
3.2 Partitioning	<channel substructure definition> is not supported in generated code.	SDL to C Compiler
	In substructures with both process definitions and block definitions, the process definitions will be ignored.	Analyzer, SDL to C Compiler
3.3 Refinement	<signal refinement> is not supported.	SDL to C Compiler
4.2 Macro	<macro definition>: macros and macro calls are transformed to PR and expanded in PR form, which restricts the usability of macros.	Analyzer
	<any area>: graphical macros can only contain flow symbols in one single flow.	SDL Editor, GR to PR converter
	<macro call>: graphical macro calls are not allowed in a higher scope than processes.	SDL Editor, GR to PR converter
4.3.3 Optional definition	<select definition>: textual optional definitions are supported, but not graphical. Any (external) synonyms used in the select expression must be defined in the system definition which may not be type based.	SDL Editor, Analyzer, SDL to C Compiler

Z.100 Chapter	Concept and Restriction	Affected Tools
4.3.4 Optional transition string	<alternative question>: question part in transition option must not contain external synonyms.	Analyzer, SDL to C Compiler
5.2.3 Axioms	<axioms>: not supported.	Analyzer, SDL to C Compiler
5.2.4 Conditional equations	<conditional equation>: not supported.	Analyzer, SDL to C Compiler
5.3.1.14 Name class literals	<name class literal>: not supported.	Analyzer, SDL to C Compiler
5.3.1.15 Literal mapping	<literal mapping>: not supported.	Analyzer, SDL to C Compiler
5.4.4.5 Timer active expression	<timer active expression>: cannot be used for timers with parameter.	SDL to C Compiler
6.2 Context parameter	<formal context parameters> and <actual context parameters> are not supported.	Analyzer, SDL to C Compiler

UML and MSC Restrictions

This section lists the major UML and MSC restrictions in the SDL Suite, with respect to the OMG specification for UML, and the ITU-T recommendation for MSC.

References below are to sections in the OMG document “ad/99-06-08” (UML Notation Guide, version 1.3, June 1999), and the Z.120 recommendation.

UML Class Diagrams

The following features of UML 1.3 Static Structure (Class) diagrams are not supported:

- Packages (section 3.13), but package references are allowed in class names
- Interfaces (section 3.28)
- Parametrized class, or template (section 3.29)
- Composite objects (section 3.39)

- Xor-associations (section 3.41.5)
- Navigability in associations (section 3.42.2)
- N-ary associations (section 3.46)
- Constraints in generalizations (section 3.49.2)
- Dependency, i.e. dashed arrows (section 3.50)

UML Statechart Diagrams

The following features of UML 1.3 Statechart diagrams are not supported:

- Hidden decomposition indicator icon (section 3.76)
- Substate compartment with concurrent states (section 3.76)
- Transition times (section 3.78)
- Concurrent transitions (section 3.79)
- History state indicator (section 3.80)
- Stubbed transitions (section 3.80)
- Factored transition paths (section 3.81)
- Submachine states (section 3.82)
- Synch states (section 3.83)

UML Implementation Diagrams

The notation used in the DP Editor diagrams is based on UML 1.3 Implementation diagrams. The major differences are the following:

- No dashed-arrow dependencies (on nodes and components)
- No interfaces (on components)
- No migration (of components from node to node or of objects from component to component)
- No graphical nesting (only composite aggregation) to show containment
- The thread concept has been added

MSC and HMSC

- General ordering for events is not supported (Z.120 chapter 4.5).
- Gates are not supported (Z.120 chapter 4.4).
- Comment symbols in HMSC diagrams are not supported.
- Parallel frames in HMSC diagrams are not supported (Z.120 chapter 5.5).

TTCN Restrictions

- The TTCN language supported in the TTCN Suite is according to an interim version of the new TTCN standard labeled Delivery 8.3, August 2, 1996, with the exception of the Modular TTCN features that are according to the Delivery 9.6 version.
- Due to the Delivery 9.6 Modular TTCN features, the support for the TTCN Package document type has been dropped.
- The import/export mechanism of the Modular TTCN feature is currently not able to handle import that uses the "xxx[yyy]" construct needed for importing named numbers, and neither able to handle the "xxx::yyy" construct for importing objects with name collisions.
- More general restrictions and modifications to the TTCN and ASN.1 support in the TTCN Suite is described in chapter 38, *Languages Supported in the TTCN Suite, in the User's Manual*.

Tool Limitations

SDL Suite and TTCN Suite Limitations

Operating System and Windowing Environment

- When you run SDL Suite and TTCN Suite on X terminals or PCs with X server software, mouse-clicks are sometimes not transmitted quickly enough for SDL Suite and TTCN Suite to recognize it.

  The X resource `*multiClickTime` can be used to change the default threshold of 250 ms if you have a UNIX terminal with the `DISPLAY` environment variable set to your local X server. You can write:

  ```
  xrdb -merge
  *multiClickTime: 1000
  ^D
  ```

  The appropriate value (in milliseconds) may depend on your hardware/network.

On-Line Help and Documentation

- When links are followed from the index in the on-line help, the positioning in the HTML help file may not always be exact. In some cases, the positioning is made a few lines down in the describing text. In the case of tables, the positioning may be after the table.

  The workaround is to scroll the text up to find the heading, bulleted item or table row describing the corresponding topic.

- In chapter 78, *SOMT Tutorial, in the User's Manual*, it is assumed that you have basic knowledge of the Link Manager. However, this tool is not covered in the earlier tutorials. For information about the Link Manager, see "Implinks and Endpoints" on page 427 in chapter 9, *Implinks and Endpoints, in the User's Manual*.
- There are no tutorials on the State Chart and HMSC Editors.
- In the MSC Editor, the appearance of some MSC symbols and lines has been updated but all MSC diagrams depicted in the documentation have not been updated accordingly.

- All dialogs and diagrams in the documentation containing the previous release versions in paths, etc. have not been updated to reflect SDL Suite 6.3.
- If the last instance of Netscape was an application other than the browser, help files will not open. The work around is to change the preference for Help*NetscapeCommand to `netscape -noraise -remote 'openURL(URLTOOPEN,new-window)'`.

ASN.1 Utilities

The ASN.1 Utilities handle all constructs of ASN.1 as defined in ITU-T recommendation X.680 with extensibility features, X.681 and X.682. Some of the X.683 construct are also supported. See restrictions below for unsupported constructs. There is no support for features defined in the old ASN.1 version X.208.

Restrictions to X.680

ASN.1 Utilities handle all constructs defined in X.680, the following semantic concepts of X.680 are not supported:

- Modules are distinguished from each other only by module name. Object identifier following the module name in GlobalModuleReference is ignored and module name is always considered definitive.
- Values can not be assigned to EmbeddedPDVType, ExternalType and UnrestrictedCharacterStringType, these types can not be constrained.
- The contents of character strings representing GeneralizedTime and UniversalTime types is not checked.

Restrictions to X.681

- Object class references (X.681, 7.1) can contain lower-case letters. Object class reference consist of a sequence of characters as specified for a type reference.
- Lower-case letters and digits can be included to Word item (X.681, 7.9). Word consists of a sequence of characters as specified for a type reference.

Restrictions to X.682

- At-notation is not checked when assigning values to the types with component relations.
- Multiple table constraints on one type are not allowed, for example “MY-CLASS.&TypeField ({MY-SET1}) ({MY-SET2})” is not allowed.

Restrictions to X.683

- Parameterized object classes, objects and object sets and value set types are not supported; only parameterized types and values can be specified, though all of the above entities can be used as parameters themselves.
- Dummy governors are not supported.

Restrictions to X.690 and X.691

- Only one reference to identifier component is supported by ASN.1 coders. For the components restricted by two or more component relations in the AtNotationList (see X.681, 10.7), only the first restriction will be handled by the ASN.1 coders.
- X.690 8.11.3 restriction. The order of data values in a set value decoding restricted by canonical order (X.680 8.4) as default compile configuration. Use CODER_BER_CANONICAL_OFF compile switch to avoid of this restriction.

- The following ASN.1 types defined in X.680 are NOT supported by ASN.1 encoders and decoders:
  - BMPString
  - GeneralString
  - GraphicString
  - ISO646String
  - TeletexString
  - T61String
  - UniversalString
  - UTF8String
  - VideotexString
  - UnrestrictedCharacterStringType
  - EmbeddedPDVType
  - ExternalType
  - InstanceOfType
  - ObjectDescriptor

Restrictions to Z.105

The following concepts of Z.105 are not supported by the ASN.1 Utilities:

- ASN.1 definitions can not be put in text symbols in SDL diagrams; they can only be placed in separate ASN.1 modules, whereas Z.105 allows free mixing of SDL and ASN.1 definitions.
- Value notations for types defined in another module are not supported. For example; `val X ::= 3` is only allowed if `X` is defined in the same module. External value references should be used instead.
- Use of the ASN.1 value notation within SDL expressions is not supported for bit string, octet string, set, set of, sequence, and sequence of.

- Value notation for SEQUENCE/SET where optional/default components have been assigned a value cannot be mapped to SDL unless SDL optional support in make operator is switched ON.
- UniversalString, GraphicString, and EXTERNAL are translated to SDL types with the same name, but no implementation for these types is available (have to be provided by you). The reason is that IBM Rational considers the mapping in Z.105 as insufficient (does not really allow extended character sets).
- There are some restrictions to use of type REAL. REAL values can be used in constraints, not however REAL values in value range. Example:

```
SDL_REAL1 ::= REAL ( 0 )                                    -- OK
SDL_REAL2 ::= REAL ( { mantissa 2, base 10, exponent 1 } ) -- OK
SDL_REAL3 ::= REAL ( 0 .. PLUS-INFINITY )                   -- FAIL
SDL_REAL4 ::= REAL ( MINUS-INFINITY .. v4 )                 -- FAIL
```

- PLUS-INFINITY and MINUS-INFINITY are translated to some large numbers. If these are not large enough for a specific platform/application, you must define these SDL synonyms yourself.
- If there are name clashes between named bits, named numbers or ASN.1 values, only the first occurrence of the synonym will be generated to SDL, for example, for ASN.1 definitions `T1 ::= BIT STRING { a(3) },b INTEGER ::= 11` and `T2 ::= INTEGER { a(4), b(5) }` only synonyms `a` from BIT STRING and `b` for INTEGER value will be generated to SDL with warning messages because generation of all synonyms according to the Z.105 standard will result in an SDL analyzer error.
- In SDL (Analyzer) literals and synonyms with the same name will clash if they have the same name. So, for the following ASN.1 definitions `T ::= ENUMERATED {x, y}     x INTEGER ::= 15 val T ::= x` generated SDL `newtype T literals x,y operators ordering; endnewtype; synonym x Integer = 15; synonym val T = x;` will not compile: *`ERROR 375 Type mismatch for synonym or literal: synonym val T = ? x INFO 352 x is one of the visible definitions`*. Although a simple workaround for this problem is to add a qualifier before value `x` to the generated SDL : `synonym val T = <<type T>>x; .`

Print

- When printing to EPS files (one per page) the files will be named after the page in a case sensitive way. On Windows, when pages have the same name only differed by case, only one file will be printed containing the last page, due to the limitation in Windows concerning case.
- Colors are not supported for FrameMaker or Interleaf printing.

SDL Suite Tool Limitations

All Editors

- Diagram names may not exceed 255 characters.

MSC Editor

- The inline expressions always cover all instance axis. The reason for this limitation is that gates (and specifically inline gate interfaces) are not supported.
- The column-form of instances and coregions is not supported, only the line-form.
- A syntax error occurs when logging to MSC if an SDL page name contains two consecutive underscores, like “pg__name”. Error message should begin, ‘`MSCE: Service INSERTOBJECT failed.`’

OM Editor

- Use *Browse&Edit Class* dialog with care if the symbol contains syntax errors.

  The syntax checker on attributes and operations have error recovery functionality, and can sometimes “recover” in an unwanted way. Use the Browse&Edit Class dialog for **editing** purposes only with special care if a syntax error is detected. In the drawing area, syntax errors are highlighted with a red underlining in the symbol’s corresponding text compartments. The browsing functionality of the Browse&Edit Class dialog can however be used without considerations even if syntax errors are detected.

UML2SDL utility

- Currently there is no special recognition of the use of an SDL predefined type in the UML diagrams. Thus when having an attribute of type Pid in the UML Suite an (erroneous) newtype definition for the type Pid will be generated.

Text Editor

- When you edit large portions of text containing many endpoints, updates can become slow in the Text Editor.

  As a work-around, turn off endpoint display if you encounter performance problems.

- On Windows, when using double-click to select words and then typing something to replace the selected word, a mismatch will occur between the text shown in the window and the actual text handled internally in the editor. The space shown after the new word is not recognized internally and is therefore missing when the text is saved and later reopened. The same happens if any text selected ends by the newline character. When replacing the selected characters with new text the ending newline will be gone when the text is saved.

Chinese characters can not be used using an English Windows installation. By using an English Windows installation with Chinese language support it is possible to enter Chinese characters in the Text Editor. However when the textfile is reopened the original text has been modified and are not displayed correctly.

Organizer

- On the Japanese version only ASCII input is supported when using the Quick Search functionality (pressing Ctrl + f).

SDL Editor

- Diagram names may not exceed 255 characters.
- If the colors red and blue are used for some symbols, you may find it hard to interpret the graphical view resulting from the Diff operation, as these colors are used for highlighting the differences between two diagrams. To avoid this problem, do not use the colors red and blue on symbols when you compare diagrams.

- The special graphical symbols for internal input and output are not available, as the use of them is explicitly discouraged in Z.100. The same behavior can be achieved with the normal input and output symbols.
- Graphical macros (macro diagrams) can only contain flow symbols in one single flow, without branches. Graphical macro calls are not allowed in a higher scope than process (system, block, etc.).
- The option symbol (dashed polygon) is not available.
- Nested diagrams are not supported, that is, diagrams must contain reference symbols to diagrams at lower levels. For instance, a system with a block must be drawn as two diagrams, where the system diagram contains a block reference symbol to the block diagram.
- When the preference *Editor*AlwaysNewWindow* is used and for example the search command is used, new windows can be shown. As long as the dialog are still opened the menu bar for the new window should be dimmed but some commands are available. Choosing these commands should be avoided when the dialog is shown.

Print

- The FrameMaker or Interleaf integration do not support colors.
- It is only possible to change the font used to print text files by using the preference SDT*PrintFontFamily.
- When printing to html, diagram page names containing '\' are not supported.

Emacs Integration (UNIX only)

- To function properly, the SDL Suite assumes GNU Emacs 19.31 or later.
- Undo is not supported for the link handling commands.
- If Emacs is started from SDL Suite and TTCN Suite and SDL Suite and TTCN Suite is shut down, but not Emacs, the Emacs session cannot be re-connected to another SDL Suite and TTCN Suite session.

Microsoft Word Integration (Windows only)

- Microsoft Word Integration does not work with MS Word 2003. The installation will fail with error 6068, Programmatic access to Visual Basic Project is not trusted.
- When a MS Word document becomes modified, its icon is not marked as “modified” in the Organizer.
- A new MS Word document is always named “Document<x>”, where <x> is an integer, independently of what name it was given in the Organizer.
- Undo is not supported for the link handling commands.
- Endpoints in MS Word documents sometimes need to be synchronized with the Link Manager manually: Use the *Update* menu choice.
- In the Organizer, multiple instances of MS Word cannot be handled.
- Due to problems with recognizing read/write security permissions on NTFS files/directories, some odd behavior in the communication with MS Word may occur. This shows up as two error boxes with almost the same contents, one in MS Word and one in the Organizer. In the Organizer, the access rights on such files are incorrectly marked as “rw”.
- Since part of the implementation is done with MS Word macros, conflicts may arise when combined with other templates/macros. This must be considered on a case-to-case basis and possibly the different macros may have to be merged. The implementation does not work with for example “Macro Virus Protection Tool” from Microsoft.
- If MS Word is started from SDL Suite and TTCN Suite and SDL Suite and TTCN Suite is shut down, but not MS Word, the MS Word session cannot be re-connected to another SDL Suite and TTCN Suite session.

SDL Analyzer

Generic functions

The default way of generating code for SDL operators was changed in SDL Suite 4.2. This method replaces generated functions for assignments, equal test, free and predefined operators in generators with generic functions.

This means that parameter passing has changed. Parameters are passed as addresses not as values, and data structures describing the data types are needed.

This method is backward incompatible with SDL systems that uses inline C code. The function calls must be manually changed.

Implementation Restrictions

The present version of the Analyzer is an SDL-92 analyzer. The restrictions and the implementation limits are described in this section. When possible, a work-around is described.

- Macros and macro calls are transformed to PR and expanded in PR form. This restricts the usability of macros, for instance the number of inlets and outlets must be 0 or 1. A common case that fails is a macro with states that is called from a branch of a decision. It fails because the GR to PR converter does not know the contents of the macro and treats the call like a task symbol. Other things not supported because of this, are variable declarations and type definitions.
- The Analyzer assumes the layout of external properties to follow the example below. That is, each external formalism name should be on the same line as the alternative keyword and the external data descriptions should be on lines of their own.

Example 1

```
alternative a,b;
   abc abc t5;
   abc abc t5-2;
endalternative;
b;
```

The PR to GR Converter

- The PR system will be changed to have all its definitions referenced to match the way the SDL Suite handles diagrams. A referenced definition needs a qualifier if name and type are not unique in the system. Such qualifiers are not supplied automatically: use the *Update Heading* command in the Organizer.

- Comments in PR files `/* comment text */` may sometimes be skipped during conversion into GR format.

Restrictions on Syntactic Analysis

- Check is not done for task symbol with only comments. As a consequence, you can get compile errors when having task symbols that only holds comments (`/* ... */`) in them.
- The Analyzer cannot handle spaces in names (a space is a shorthand for an underline). Replace spaces with explicit underlines.

Restrictions on Semantic Analysis

- The semantics of virtual types is unclear.
- Gate constraints on implicit signal routes are not checked.
- Implicit gate connections are not checked.
- Signals of global procedures are not properly checked.
- Only limited tests on output via.
- It is not checked that instantiated types have signals in the gate constraints.
- Context parameters are not implemented.
- Conditional equations are not implemented.
- Some checks may be lost in nested virtual definitions.
- Package interfaces and definition selection list in USE statements are not implemented.
- Remote procedure calls are not checked against imported procedure specifications.
- Imported procedure specifications are not checked against exported procedures.
- Import expressions are not checked against imported variable specifications.

- Imported variables are not checked against exporting processes.
- The check that substructure name in subtype and supertype must be the same is not implemented, since it would make the use of “block in block” useless for inheritance.
- Operator definitions are not checked against operator signatures.
- Name class literals in user defined data types cannot be handled in the Semantic Analyzer. A nameclass literal is a shorthand for writing a (possibly infinite) set of literal names defined by a regular expression. List the literals explicitly or use inheritance from the predefined data types.
- Question and answers in decisions are always considered formal, that is, informal text is considered to be a character string.
- Inheritance from data type definitions containing qualifiers referencing that type definition is not allowed, since qualifiers are copied and not changed during the expansion of inheritance.
- When analyzing part of a system, checks depending on the omitted parts will not be performed.
- When a substructure contains both processes and blocks, it should be possible to select which implementation that is preferred. This is not possible; the process definitions will not be analyzed.
- Signal refinement is not supported.

Semantic Checks that are not Performed

Analysis of the following semantic SDL rules is not performed:

- The consistent partitioning subset must be a consistent refinement subset.
- The decision answers must be mutually exclusive. This is not tested for all possible data types.
- An exported variable must exist for each imported variable.
- The type check and the evaluation and check of the equivalence classes in axioms is not performed.

- Generator names and generator formal parameters must not be used in a qualifier, qualified, followed by an exclamation, or used in a default assignment.

Cadvanced SDL to C Compiler

- If you have a block type or a process type, with user defined separation; all other block/process types that do inherit from that block/process type also needs to have user defined separations. If they do not, the order in which the h files are included can get mixed up.
- Compilation Error if **user defined separation** and XFREESIGNALFUNCS defined. The signal s_c2b_1 is defined in the file BlockB.h since user defined separation is used. The definition of `cm_Free_Signal` in the system.c file uses the signal s_c2b_1 which is not known since it is defined in a file which is not included.
- Channel substructures are not allowed.
- Signal refinements are not supported.
- Context parameters are not supported.
- Timer active expressions for timers with parameters are not allowed.
- In generated code there will be no difference between a channel with delay and a channel with NODELAY.
- Integer and Real are restricted in range and precision, as they are implemented using the C int (32 bit integers) and double types. Charstrings may not contain the character NUL, as Charstring is implemented using `char *` in C, and `char *` is terminated by NUL (=0).
- Axioms and literal mappings may be part of an abstract data type definition, but that information is not used in any way by the SDL to C Compiler.
- In abstract data types, the following concepts cannot be handled:
  - Name class literals
  - Naming a literal using a Charstring literal
- A sort definition may not refer to itself, directly or indirectly.

- A component of a struct will, in the generated code, have the same name as in SDL. Characters that are not letters, numerals or underscore characters will, however, be removed. The remaining part of the name must be a valid C identifier to be accepted by the C compiler, that is:
  - It should start with a letter.
  - It should not coincide with a C reserved word.
  - It should be unique within the C struct representing the struct.
- Multiple paths from a decision for a certain decision expression value are not checked. A simulation program will simply choose one of the existing paths.
- Overflow of integer and real values are checked at the C level if the actual C system performs these checks.
- A process that exports procedures cannot call (directly or indirectly) a global procedure containing states.
- Output VIA ALL C, where C leads to a block instance set, is not handled correctly. One signal should be sent to each block instance, but only a single signal will be sent.
- The `.ifc` file contains `#define` definitions for the kind of SDL units for which this is possible. All the SDL to C Compilers follow this principle. This can, depending on the C compiler, lead to compilation errors/warnings or to runtime errors in some cases. The following example shows how the problem may occur:

```
  newtype x
   literals green, blue;
endnewtype x;

newtype y
   literals blue, green;
endnewtype y;
```

The both literals `green` and `blue` will then be created like this:

```
/* For newtype x */
#define green  0
#define blue   1

/* For newtype y */
#define blue   0
#define green  1
```

The C compiler then usually complains that the macros `blue` and `green` are redefined with a different value. Some C compilers will continue in any case and produce an executable. Runtime errors are the consequence on this, because when `green` and `blue` are used together with the first `newtype x`, the literal values from the second `newtype y` are used by the C compiler. In order to avoid this problem, a prefix for each literal should be manually added, like this:

```
newtype x
   literals x_green, x_blue;
endnewtype x;

newtype y
   literals y_blue, y_green;
endnewtype y;
```

The `green` and `blue` example is not a good one, because it is possible to avoid the problem by using the inherit construct. The literals from the `newtype x` could be inherited into the `newtype y`. More problems will occur if several entity classes are mixed up from the C compiler, like if there would for example be a signal called `green`, which is also created as a `#define`. You are strongly recommended to avoid name conflicts in the `.ifc` file by manually adding prefixes.

- any(Sort) where Sort is a syntype is only implemented if the syntype contains at most one range condition which is of the form a:b, that is one limited range. If it is a syntype of a real type, e.g. Real or Time, with a range condition it is not implemented.
- any(Real) does not generate random decimal values, only whole numbers.
- When the preference SDT*RelativeSDTREF is on, and Edit Separation is used with a separate file name, compile errors may occur requesting a file that does not exist. This can be resolved by either not using a separate name or turning the Relative SDT References off.

Targeting Expert

- When you use Targeting Expert to generate code based on a Deployment diagram the “User defined separation” information from the Organizer is disregarded.

- The name of the generated C file is different if you have a space in the path for the source SDL or not. If the path contains a space, the name will be `<systemname>.c` otherwise the name will be `component.c`.
- The automatic and manual configuration/scaling for the Cadvanced libraries will not be used per default in all pre-defined integrations. The compiler flag `-DUSER_CONFIG` must be set to use configurations.

  The solution is to change the alignment to 8 for all types in the "Host Connection" section in Targeting expert. I.e "Target data coding: alignment" should have all entries set to 8. These values are read from the file sdtmt.opt.
- There is a memory leak in function GenBag_Take2 when a SET OF is used.

  Example that leads to memory leak:

  ```
  y:= Take(test_set, 1)!item
  ```

CPP2SDL

- The following C/C++ dialects are supported by the tool:
  - ANSI C/C++
  - Microsoft Visual C/C++
  - Gnu C/C++
- The tool performs a complete syntactic analysis of the input C/C++ program, but only parts of the semantic analysis is done.
- C++ exceptions are not supported.
- C++ template declarations are not translated to SDL directly. Instead, instantiations of the templates must be provided, and these will be translated.
- The support for function pointers is limited. Function pointer types are represented in SDL (as untyped pointers), but it is for example not possible to call the functions pointed at.
- Expressions containing usages of the `sizeof` operator are not always correctly evaluated.
- C++ cast expressions are not fully supported.

- Members inherited from a virtual base class are sometimes not accessible in the SDL translation.
- Classes that inherit from multiple template instantiations of the same template are not fully supported.
- "sizeof" is not correctly handled under certain conditions cause problems.
- There can be problems when using anonymous types for example:

```
typedef struct {
  int i;
  float f;
}mess;
```

instead it should be:

```
typedef struct mess {
  int i;
  float f;
}mess;
```

SDL Simulator

- Use of the add-macro feature in combination with passing an array or aggregate as an actual parameter can in some cases cause the execution to lose or misinterpret parameter information. Parameters are not accurately passed to input scripts. This is also the case when passing multiple characters (enclosed in single quotation marks). A possible workaround is to parameterize any input script by passing explicit parameters rather than to use macros.
- There is a limitation in the simulators way of handling qualifiers. This results in that the analyzer may not detect that two instances of blocks have the same name. As a consequence it is possible to build a simulator executable without problem, and it may for example not be possible to inspect variables due to ambiguous name on the instances.
- If you have already generated an SDL simulator from the Organizer, and want to generate a new one with other options or with another selection, you should perform a Full Make, as changes in options or selection is not handled by the build process. Otherwise compilation or link errors might be the result of the build process.

- There are no range checks implemented for general arrays. General arrays are implemented as linked-lists because the code generator can not determine whether or not it can be implemented as an ordinary array. This occurs if a synonym translated to a variable is used in a range condition of a syntype and the syntype is used as an index sort in an array or powerset instantiation. The reason for this is that the length of the array cannot depend on a variable in C.

  An example:

  ```
  SYNONYM Lo NATURAL = 1;
  SYNONYM Hi NATURAL = 10;
  SYNTYPE BufferType = NATURAL
  CONSTANTS Lo , Hi
  ENDSYNTYPE;
  NEWTYPE MessageBufferType
    Array( BufferType, CharString)
  ENDNEWTYPE;
  ```

SDL Explorer/Autolink

- When using the Verify MSC command the MSC file cannot have spaces in the filename or pathname.
- Bitfields in struct that also contains pointers are not handled.
- C unions are not allowed to contain pointers.
- Encoding/Decoding functionality cannot be used together with the SDL Explorer kernel.
- Validation of MSCs containing a signal sending a parameter of type ptr_char, or Charstar will give the following error: “No reference to dynamically allocated memory.”

  The RefError occurs because when you read in the MSC diagram there is a parameter of type Charstar and reading this will create the dynamic memory which can not be handled correctly.

Examples

- **On UNIX**, the `phone` example cannot be used together with the graphical SimUI if the Phone application is compiled and linked with the ApplicationDebug kernel.

Real-Time Operating System Support

Light Integration

The same restrictions as for the SDL to C Compiler apply. See “Cadvanced SDL to C Compiler” on page 24.

Threaded Integration

The same restrictions as for the SDL to C Compiler apply. See “Cadvanced SDL to C Compiler” on page 24

- Threaded Integration can only be generated with the Deployment Editor and Targeting Expert i.e. the make feature in the Organizer can NOT be used.
- The Threaded Integration for VxWorks uses POSIX library functions i.e. the VxWorks kernel must be compiled with the POSIX libraries.
- In a Threaded Integration, the xInEnv function is invoked when a state transition occurs in the SDL system. This means that intervals between the xInEnv calls are irregular and depend on the properties of the running SDL system. When writing external code for input into an SDL system, it is recommended that the code is run in a thread of its own, using xGetSignal and SDL_Output for inserting signals into the system.
- On-line MSC trace under VxWorks softkernel does not work.

TCP/IP Communication Module

- On VxWorks, the TCP/IP communication module has been tested on a softkernel running on Solaris. In order to connect to a Vxworks softkernel application, the loopback IP address of the host machine must be used. In most cases, this IP address is “127.0.1.0”.
- On OSE, the TCP/IP communication module has been tested on a softkernel running on Solaris. In order to connect to a remote component from an OSE softkernel application, the IP address of the remote host must be given explicitly. A regular hostname, e.g. “the_computer.the_company.com”, will not suffice.

Tight Integration

- The tight integrations in the SDL Suite are only applicable for Cadvanced – not for Cmicro.
- Only Solaris (Posix 4), Win32, VxWorks, and OSE delta integrations are available as downloads.
- Priority input is not handled.
- Continuous signals are not handled.
- Services are not handled.
- An RPC call of value returning procedure without parameters does not work.
- The SDL `ANY` construct is not properly implemented.
- The timer set construct `SET(5,T1)` is not supported. Instead use `SET(NOW+5,T1)`
- ADT Library files located in `$telelogic/sdt/include/ADT`:
  - file.pr:

    Should work if supported in the RTOS environment. It might be necessary to enclose the function `xGetValue` with the following code to be able to compile with some compilers.

```
#ifdef XMONITOR
...
#endif
```

  - list1.pr and list2.pr:

    Should not be used since they are implemented with dynamic pointers which could lead to memory leaks.
  - byte.pr:

    Works OK.
  - unsigned.pr, unsigned_long.pr, longint.pr:

    Works OK.
  - pidlist.pr:

Works OK but only with the version supplied in the RTOS delivery:
`$telelogic/sdt/sdtdir/RTOS/INCLUDE/pidlist.pr`.

– idnode.pr:

The functions `FirstPId` and `NoOfProcesses` may work under some conditions, the rest of the functions do not.

– pointer.pr:

Should not be used since there is a big risk of memory leaks.

- Support for Kernighan & Ritchie C is discontinued.
- The integration code does not check the upper limit for the number of process instances as declared in the SDL system.
- The same data type is used to represent TIME and DURATION; in most operating system this is an unsigned integer. Thus TIME or DURATION expressions may not evaluate to negative values. This also applies to intermediate expression results.
- The SDL error term is not supported in Cadvanced Tight Integrations. E.g. “return error;” can not be used in Cadvanced Tight Integrations.
- REVEAL and VIEW are not supported. Applications using this concept may crash or behave in an undefined way due to concurrency problems.
- EXPORT and IMPORT are not supported. Applications using this concept may crash or behave in an undefined way due to concurrency problems.
- In addition, the same restrictions as for the SDL to C Compiler also apply. See “Cadvanced SDL to C Compiler” on page 24.

SDL C Compiler Driver (SCCD)

- RESTRICTION: SCCD under DOS/Windows does not handle include paths in double quotes.

Cmicro SDL to C Compiler

- The names of processes in different blocks must not be the same. The reason for this restriction is the missing symbol tree in Cmicro.

It is impossible to address for example two processes with the same name in the same C module (in the SDL environment). As a workaround, you should always have different names for the different processes in the system. You can easily accomplished this by adding a prefix to each process name.

- Predefined sort carray:

  It is impossible to return variables of the predefined sort carray from an SDL procedure because the ANSI C language cannot handle this. It is however possible to enclose the carray into a newtype struct value as a workaround.

- Timers with parameters and the preemptive Cmicro Kernel do not work together.
- If a procedure returns a value of a sort that is passed as address, an assignment like `call prd(invar, retvar);` is not supported. Instead use `retvar := call prd(invar);` .
- You shouldn't have syntypes named as the following:

```
int
unsigned_int
long_int
unsigned_long_int
short_int
unsigned_short_int
char
signed_char
unsigned_char
float
double
ptr_void
bool
wchar_t
```

Cextreme SDL to C Compiler

The following are not supported in the Cextreme SDL to C compiler:

- Services
- Procedures with states
- Access to outer procedure dcl or fpar
- View / Reveal

- Continuous signal
- Priority input and output
- Priority for process (#PRIO) for processes
- Channel substructure
- Timers with more than one parameter
- Timers with another parameter than sort integer
- Remote procedure call in procedure or operator
- Use of qualifier when calling a virtual procedure

SDL Target Tester

- When the target system is suspended (by sending the suspend command), any signal that is sent to the target will be discarded.
- When, with the Output-PAR/NPAR command, you want to send a signal to the target system, a list with all the signals that are defined in the system is presented. You should keep in mind that you always have to address the right process instance when specifying the parameters of the SDL Target Tester's output command.
- It is not possible to send charstrings or pointers with the Output-PAR command to the target.
- Several compilers, e.g. the Tasking 80166, are giving incorrect information about the message length.

  This error is detected by the SDL Target Tester and displayed once. The trace of SDL events will be correct as only the CRC check is not involved in the following communication.
- The SDL Target Tester should only be started from the Targeting Expert's *Tools* menu since there may be problems when the SDL Target Tester is invoked standalone.
- Page file and convert file in SDL Target Tester does not work.
- **On Unix**: The Cmicro Postmaster leaves temporary files below `/tmp`. You should remove these files from time to time by hand (`rm /tmp/.sdt.cmicro.*`)

- It is impossible to specify a breakpoint on a timer input.
- If the debug mode of Cmicro Postmaster is used, there might be too long messages, in which case the Postmaster will crash with a core dump.
- If the line numbering in the SDL Target Tester is switched on and off again, sometimes the first columns in the text area are split. As a work around and to prevent this, line numbering always should be switched on at the beginning and never switched off again.
- The context sensitive help sometimes indicates that a command may be used, though the command cannot be used in current context (setting a breakpoint is allowed although no target system is connected and running).
- **On UNIX**: If the `sdtmt.btn` file contains too many entries that are marked as ":MENU", the SDL Target Tester is not able to scale its window size when it is invoked. If you resize the window manually, the size will be correct.
- The Parametertrace is restricted to Predefined Sorts, Syntypes, Enums, Newtype Structs and Newtype Array.
  If the SDL Target Tester encounters an unknown data type, the rest of the parameters is shown as hex-buffer.
  Limitations:
  It is not possible to show the contents of a Charstring.
  If you use wrong values for `LENGTH_`, `ALIGN_` and `ENDIAN_` in the file `sdtmt.opt` it is possible that the Tester decodes all parameters, but the shown values are wrong.
- Searching the Text trace in the Target Tester with regular expressions is disabled by default. The Tester crashes if reaching the end of the trace, i.e. there is no more regular expression found in the trace.
  The feature can be enabled by setting the environment variable TT_REGEXP to 1 before starting the Target Tester.

Application Generation with Cmicro

- Not all possible errors are checked by the Targeting Expert. In most cases, a compilation error is the result you will get from an incorrect configuration. The on-line help will support you to configure correctly.

- If you try to `#include "systemname.ifc"` in `ml_typ.h`, or for some other reason, there may be further naming conflicts, for example during preprocessing:

  ```
  sctpred.c: In function `yAss_SDL_Object_Identifier':
       ml_pred.c:2109: parse error before `0'
       ml_pred.c:2109: parse error before `0'
  ```

  The reason for the error above (and other behavior that is impossible to foresee) was that there was a signal named "`SIn`". This will be generated as a `#define SIn 0`. In the `sctpred.c` file, there is a variable called `SIn` (in the macro `yAss_SDL_Object_Identifier`). The C preprocessor then replaces the variable name with the `#define` value of `SIn`, which ended up in the parse error above.

  In order to avoid that, you should never include code generated header or `.ifc` files in any part of the Cmicro Library itself or in the generated C code. The `.ifc` file should only be included in the SDL environment files (like `env.c`).

- In the generated C code, the following warning is repeated for each variable that is unused:

  ```
  "Unused Variable : <variablename>"
  ```

  These warnings in generated C code are not that easy to remove because it is difficult to calculate this during code generation. These warnings should not produce real problems (except that it is sometimes impossible to read the large C compiler output). The following variables are sometimes referred:

  ```
           warning: unused variable `yOutputSignal'
           warning: unused variable `yVarP'
           warning: unused variable `ySVarP'
  ```

  The same warning as above is thrown from C compilers when signals without parameters are used.

  This warning is accepted in the `mk_outp.c` module:

  ```
  mk_outp.c:warning: unused variable `P_Parameter'
  ```

  A similar warning (accepted also) is within `mk_stim.c`:

  ```
  mk_stim.c:warning: `SystemTime' defined but not used
  ```

- Warnings like: “Control reaches end of non-void function”

  As the SDL to C Compiler basically creates a `switch()` construct in C, and every return from the process activity description function (PAD function) occurs in all the case branches of that switch, the PAD function contains no return statement after the switch. This is the reason why some C compilers complain that the “Control reaches end of non-void function”. This is an accepted warning. It is possible for you to remove that warning in the `ml_typ.h` or `user_cc.h` file by modifying the `END_PAD` macro like this:

  ```
  #define END_PAD(PrsNameWithPrefix) return (SDL_DASH_NEXTSTATE);
  ```

- Warnings: missing braces around initializer for `xSTATE_INDEX_zyyyy_xxxxxxx[0]`

  This is an accepted error so far, because each known C compiler handles this correctly. The reason why the SDL to C Compiler generates like this is that is still should be conform to not just only ANSI C, but also Kernighan & Ritchie C.

- Some C compilers fail in preprocessing of very long macro definitions (like all the C Compilers available on the market for the 8051 family of CPU's). Such long macro calls occur in the predefined generators (String, Powerset, Bag).

  If the C compiler (or specifically the C preprocessor) complains about a macro definition that is too long, or produces erroneous output, it is not possible to use these kind of generators. The macro `X_LONG_MACROS` should be set with the Targeting Expert.

- It could also be impossible for some C compilers for embedded systems to compile recursive macro calls. Recursive macro calls appear for SDL statements like for example:

  ```
          a:=b and c and d;
  ```

  If such a problem occurs, it is strongly recommended to avoid such statements for example with the following work around:

  ```
         tmp:=b and c, a := tmp and d;
  ```

- When it comes to target applications, you always have to specify how memory is allocated (see C module `mk_cpu.c` in the template directory) and what should happen if there is no more memory available. As the template C files does not know what you want to do, there is a C macro called NO_MORE_MEMORY that you must define. Any reaction on such a fatal error might be appropriate, like “Hang up the program”, or “Reset”, or “Print out an error message”. Printing out error messages may however fail as well, if the print function allocates memory.

- The default values for variables are not set in a procedure when XMK_USE_KERNEL_INIT is defined. This is a problem which cannot be solved easily.

  There are two work-arounds:

  – Do not define XMK_USE_KERNEL_INIT

  – Do not use default values for variables in a procedure, e.g. this is forbidden in a procedure (in a process it does work very well):

  ```
  DCL a integer;
  DCL b integer;
     task b:=a+b; /* non deterministic results */
  ```

  This is better:

  ```
  DCL a integer = 0;
  DCL b integer = 0;
     task b:=a+b; /* deterministic result ! */
  ```

Use of ADTs from the SDL Suite in Cmicro

The packages and ADTs delivered with the SDL Suite are mainly to be used in Cbasic/Cadvanced applications. Some of these ADTs may however also be useful for Cmicro applications. Others cannot be used together with Cmicro as they contain references to C code from Cbasic/Cadvanced. A list of restrictions and recommendations can be found in “Abstract Data Types” on page 3408 in chapter 65, *The Cmicro SDL to C Compiler, in the User´s Manual*.

Combining Cadvanced / Cmicro Code

Mixing C code from different SDL to C compilers is not possible as they use their own runtime model and runtime data structures. Trying to mix up the C code will sooner or later lead to compilation errors. This restriction is true for any kind of combination of C code, including sdth2sdl.

TTCN Suite Tool Limitations

Color Problems on Solaris

- On computers where the display hardware limits the number of concurrent displayed colors, the TTCN Suite might run out of colors. It will in this case start to use what appears to be random colors. This should have no effect on the usability of the TTCN Suite, but can be somewhat annoying.

Table Editor on Solaris

- The Table Editor can only display up to approximately 1000 rows. This limitation is only present in the Table Editor, so conversion to or from MP, printing, analysis, etc. is not affected.

Analyzer

- The types BMPString and UniversalString are not supported.
- The Analyzer does not support encoding/decoding variations.
- ASN.1 enumerated type identifiers are allowed to start with both upper and lower case characters. This applies only for the TTCN Suite tool and not for the ASN.1 utilities tool.
- Defect 4980: The analyzer will make an incomplete analysis when there is an empty behavior line following a (possibly erroneous) send or receive statement.
- Type lookup:
  Due to how the type lookup mechanism is implemented, the context where a type is referenced have to be seen before encountering the value to look up.
  This has the effect that constructs like:
  `[{a 2} = variable_with_sequence_as_type]`
  is going to fail looking up the type of the sequence value.
  Work-around: Rewrite that construct to reference the variable first:
  `[variable_with_sequence_type = {a 2}]`
- When checking the type restrictions, INTEGER calculations are limited to have result in range -(2^64-1)..(2^64-1). When overflow occurs, a warning is reported, and restriction is not checked.

External ASN.1 Reference

The definition of the ASN.1 type that is referred is limited:

- References of the form `Module.Type` are not allowed in the definition.
- References to types/values in the same or other ASN.1 module is not allowed in the definition, unless separately defined.
- The limitations for the types and values referenced from TTCN also include those described in "ASN.1 Utilities" on page 13.
- When using ASN.1 PDU Type Definitions By Reference, you cannot employ BER or PER ASN.1 encoding/decoding if you are mapping the PDU names to other names local to the test suite. (The PDU names will not be recognized by the runtime system.)

TTCN ASN.1 BER Encoding/Decoding

- The following composite ASN.1 types are not supported:
  - EXTERNAL
  - EMBEDDED PDV
  - CHARACTER STRING
  - ObjectDescriptor
  - UTCTime
  - GeneralizedTime
  - TeletexString
  - VideotexString
  - GraphicString
  - GeneralString
  - T61String
  - ISO646String
  - BMPString
  - UniversalString
  - UTF8String (from X.680/1997)

  It is not possible to Encode/Decode values of these types.

The TTCN to C Compiler

The language supported by the TTCN to C compiler is a subset of the language covered by the Analyzer. This means that in some cases, constructions that have been analyzed correctly are not supported in the code generation phase. The following list describes the known limitations in the current TTCN to C compiler.

- The `INTEGER` data type is restricted in range and precision, as it is implemented using the C unsigned int (usually 32 bit) data type plus sign attribute.
- The `MIN/MAX` and the inclusive/exclusive (that is `<`) syntax in ASN.1 value range sub types are not supported.
- `INFINITY`, in all its uses, is not supported.
- Selection type and their values are not supported.
- Values of the `REAL` data type is not supported.
- Encoding/decoding variations are not supported.
- If two identifiers in the TTCN differ by a final letter, D, TestCase (Test Suite Variable and TestCaseD) then TTCN Suite could generate code with name conflicts. Example:

  ```
  Name of Test Suite Variable: TestCase
  ```

  ```
  Name of Test Case: TestCaseD
  ```

MSC Logging

The MSC generation part of an ETS is limited in scope and applicability to some particular applications. This list of limitations may be circumvented by manually editing the code in the files `mscgen.h`, `mscgen.c`, `static.c` and `gci.c`. Such edits are not supported by IBM Rational, though suggestions of improvements are welcome.

- Distributed concurrent test component logs are not supported.

  The reason for this is that it would require additional communication in-between the distributed test components. The required features for a common log of any kind are not available through the architecture of TTCN or the GCI Interface. The result of applying MSC logs for truly distributed ETSs is not defined. Note that the Composed mode may work but it will produce logs only of one component at a time.

  The internal concurrency mode of the TTCN Suite is fully supported by the MSC Generator.
- Re-start of test components

Concurrent test suites in which one given component is created more than once, will result in MSC/PR that may not be supported by the MSC Editor.

The generated MSC log will contain multiple instances with the same name, though presumably not at the same time. It is conceivable to write a text processing script which renames instances that are created multiple times. The limitation applies only to the Decomposed mode.

- Value notation limitations

  These are limitations to the value encoding for MSC/PR form. For information on how the value encoding can be completely disabled, see “Compiling an ETS with MSC Generation” on page 1332 in chapter 32, *The TTCN to C Compiler (in Windows), in the User's Manual*.

  – Some types may not be supported for the logging of values with the events of the MSC log. If they are encountered, a warning message will be appended to the event log. See the `mscgen.c:MscEncodeValue` function for details.

  – Some minor type encodings may differ from the one produced by the SDL Simulator. These differences can be fixed by editing the `mscgen.c:MscEncodeValue` function. Note that such changes are not supported by IBM Rational.

  – Some types that are not supported by the SDL Suite are handled. The HEXSTRING and derived TTCN types are not recognized by the SDL Suite.

  – The tool has a fixed size buffer for MSC value encoding. The result is that some values may be truncated (values that print to strings longer than approximately 4000 characters). See “Compiling an ETS with MSC Generation” on page 1332 in chapter 32, *The TTCN to C Compiler (in Windows), in the User's Manual* for a description of the definitions that need be changed for increasing the buffer size. If the buffer is filled up, syntactic errors in the generated MSC/PR may be the result.

  – The use of message types that have names equal to MSC keywords may result in incorrect MSCs. There is no functionality for detection of this in the MSC Generator or TTCN Analyzer.

  The MSC Editor will detect the problem if trying to read MSC/PR form files where the problem is present. It is recommended to avoid ASP, PDU and CM names that are such names. Examples of these names include “reset”, “RESET”, “stop”, “msc”, “TimeOut” etc.

- Concurrent TTCN configurations
  - The instances used in a generated MSC always include all the PCOs declared in the test suite. As a consequence, even though a PCO may not be defined to be used in a test case with a configuration, it will still be part of the generated MSC.
- The MSC Editor logging lacks some error control and may also result in excessive memory consumption by the MSC Editor in long test runs. Also, it slows down the tester significantly since it performs an extra 2 RPCs for each event that is logged. It is recommended to use this option only when developing or testing test scripts.
- It is possible to “reverse” the components used in the “composed mode” by editing the MscIn and MscOut functions such that they in the composed mode do not switch the “from” and “to” parameters. The reversed view may be conceptually more natural, though it may not be possible to process with some of the other tools of the SDL Suite and TTCN Suite family.
- There is an additional compile-time definition that is for debugging purposes: DEBUG_POST_COMMUNICATION. If set, there will be printouts to `stderr` of messages sent and received from the SDL Suite and TTCN Suite Postmaster. See also ***“POSTDEBUG” on page 494 in chapter 10, The PostMaster, in the User´s Manual***. If events seem to be missing in an MSC Editor log, these may be used to determine if there is an unhandled error condition.
- The MSC Editor has a built-in maximum number of simultaneously open diagrams. Since each test case generates a separate MSC diagram, this limit may eventually be reached when large numbers of test cases are run. The result of reaching this limit is not defined. As a consequence, it is recommended not to use MSC Editor logging when running large numbers of test cases.

TTCN Simulator

The entire list of SDL to C Compiler limitations applies also to the Simulator, see “TTCN ASN.1 BER Encoding/Decoding” on page 40.

- Monitoring of other test components can only be viewed together in the main screen log.
- Test suite variables and timers cannot be modified or viewed.
- The contents of channels cannot be modified.
- An entire test suite cannot be selected for execution. You have to do this manually, for example by selecting all test groups or all test cases.
- If you abort the execution of the SDL and TTCN Integrated Simulator when the SDL Simulator is running, further communication may be inhibited. Always restart the SDL Simulator when you abort the execution of the SDL and TTCN Integrated Simulator.

SDL and TTCN Integrated Simulator

- ASN.1 modules that have been joined/merged within the SDL system will not always work for SDL and TTCN Integrated Simulation, as the same join/merge functionality does not work the same way on the TTCN side. This means that name clashes will be solved differently on the SDL and the TTCN sides respectively.

TTCN Exerciser

Timer Duration Limitations

- If a timer has a duration exceeding approximately 2000000000 (2^31-1) time units – as defined with the timer declaration – the condition may not be noticed, and it may result in erroneous behavior.
- If a timer duration exceeds the theoretical maximum period, the condition may not be noticed, and it may result in erroneous behavior.

Value Encoding and Decoding Limitations

In general, the value encoding and decoding has been designed to be a general mechanism for reading and writing almost any value, including ones that are not part of the type system of a generated test suite. This is for being able to input totally unexpected or irregular messages. The consequence of this is also that it may be easy to “break” test suites or even the kernel by providing wildly unexpected message contents. In order to get valuable results of the simulation runs, there may be a bit of effort needed to specify the messages. While doing that, the following notes and limitations may be helpful to avoid some pitfalls:

- There is limited error checking in the value encoding and decoding, which may result in ambiguous messages being encoded or decoded, or even possibly crashes if some invalid combinations are used.

  For example, if `asp1` is an ASP (base type is sequence), then these value encodings are likely to cause a runtime error or crash if fed into the kernel:

  ```
  asp1 24
  IA5String TRUE
  ```

  This error situation is not handled, since in general the kernel is designed to allow for arbitrary objects to be input. In the general case, the situation above will be created if assigning a refined type to an object with a different base type.

- For simple types, the actual type name is not part of the encoding. This particular choice of implementation was made to improve the readability of encoded values, but it may result in ambiguities when decoding the same message.

  An example: The value `VideotexString "Hello"` is encoded as `"Hello"` only, without the actual type name. Decoding of the same message will result in an object with the default string type IA5String. In order to specify the actual type, use a prefix of the actual type, such as `VideotexString "Hello"`.

- ENUMERATED values are not encoded or decoded by name, but rather as integer values.

- CHOICE values must be possible to distinguish by value, rather than by tag since tagging is not incorporated in the encodings and decodings of the TTCN Exerciser.

- The type OBJECT IDENTIFIER is not supported.

Concurrent TTCN

- There is a built-in limit in the TTCN Exerciser to how many concurrent test components may be simultaneously active. The limit is more than 10 but less than 25. The reason for this limit is the implementation of thread scheduling in which there is a built-in overhead which grows by the number of threads. There should be no problems running up to maybe 6–8 simultaneous test components, but more may seriously degrade the performance of the kernel. To work around this for real-time testing, do consider an alternative adaptation strategy at this time (using a deterministic and fair real-time system scheduler).
- Cancelling a test case from the context of a parallel test component may lead to unexpected behavior (such as hanging or livelocked PTC threads). It is recommended to always run a concurrent test case to a `SNAPSHOT` state before cancelling.
- The result of terminating the MTC prior to all PTCs have terminated is undefined. The condition should be reported by the TTCN Exerciser, but the test result may be invalid. Terminating the MTC prior to all MTCs are terminated is illegal according to TTCN.
- There is no way to list the contents of a CP with the TTCN Exerciser.

TTCN Link

- Encoding/Decoding functionality cannot be used together with the TTCN Link kernel.
- ASN.1 modules that have been joined are not supported by TTCN Link.

Auto Link

- ASN.1 modules that have been joined are not supported by Auto Link.

Storage Format Compatibility

System Files

The file format for system files (`.sdt`) changed in SDL Suite 6.3 and TTCN Suite 6.3 as compared to 6.2. SDL Suite 6.3 and TTCN Suite 6.3 can read older system files (3.0X - 5.1), but versions earlier than 4.4 cannot read the new system files.

When saving older system files with the Organizer, a conversion to the new format is performed.

SDL/GR and MSC/GR

The storage format for SDL diagrams did change in 4.1. This means that diagrams created with 4.1 cannot be opened in previous versions. SDL Suite 4.1 can still read all previous formats.

The storage format for SDL diagrams did change in 4.0. This means that diagrams created with 4.0 cannot be opened in previous versions.

The storage format for SDL and MSC diagrams did change in SDT 3.1X as compared to SDT 3.0X (and SDT 2.X). SDT 3.1X and later versions of SDL Suite and TTCN Suite can read older files, but previous versions of SDT cannot read the new format introduced in the 3.1X version.

In addition, **the storage format for MSC diagrams was changed in version 3.5**. Version 3.5 and 3.6 of the MSC Editor can handle the same formats as version 3.4 of the editor, but version 3.4 and earlier of the MSC Editor cannot read the new MSC format.

SDL and MSC diagrams stored in SDT 2.X and SDT 3.0X format may be opened as is into SDT 3.1X and later versions of the editors (using commands such as the SDL Editor's *Open*). To preserve the original diagram structure, you must however import them into the Organizer, by using the command *Import SDL* to build a diagram structure that may be stored on a system file.

When saving the opened SDT 2.X or 3.0X diagrams on file with SDT 3.1X and later versions of the editors, a conversion to the 3.1X format is performed. Conversion to the 3.1X format may also be performed automatically when importing the diagrams into the Organizer. The option *Save imported diagrams in SDT 3.X format* is used for this purpose.

Caution!

It is not possible to open files in the SDT 3.1X format in previous versions of SDT, e.g. SDT 3.0x or SDT 2.3, or to open MSC diagrams in the new 3.5 format in previous versions of the MSC Editor. Make sure you have backup copies of your diagrams.

Generated SDL Simulators and Explorers

Simulators and Explorers should be generated with the same version of the SDL Suite as they are supposed to be executed in. They might not execute as expected when executed under a SDL Simulator Graphical User Interface of a different version. To re-generate, use the *Full Make* facility to make sure all C files are re-generated properly.

The SDL/GR trace in the SDL Editor requires the SDL diagrams to be saved in order to function properly. (The reference mechanism in SDT 3.X assumes the existence of information that is not stored in SDT 2.X files.)

Deployment Diagrams

The 4.2 Deployment Diagram file format (*.sdp) was extended. The 4.2 Deployment Editor can read and convert .sdp files created with older versions of the Deployment Editor. However, older Deployment Editor versions can not read diagrams created with this new version.

In the 4.1 release, the Deployment Diagram file format (*.sdp) was extended. The Deployment Editor in the 4.1 release can read and convert .sdp files created with older versions of the Deployment Editor. However, older Deployment Editor versions can not read Deployment Diagrams created with the 4.1 version.

Buildscript generation is no longer supported by the Deployment Editor since the 4.1 release. Buildscripts are replaced by a feature which translates Deployment Diagrams into the Partitioning Diagram Models, which are used by the Targeting Expert. This makes it possible to utilize a partitioning configuration in a Deployment Diagram when building SDL systems using the Targeting Expert.

As buildscript generation is no longer supported, build settings on node and component level are ignored when an old Deployment Diagram is converted into 4.1 format. Qualifier data on object level is automatically translated.

The TTCN Suite Database Compatibility

This section describes the different versions of the TTCN Suite database formats and how they are converted to each other. The following table shows the what formats are used in each TTCN Suite version:

TTCN Suite	Database version	Description
2.0, 2.1	0	Base version.
2.2	1	This version stores additional analyzer info.
3.0x	2	Corrected order of DefaultsRef and Configuration in TestCase.
3.1x	3	Support for Modular TTCN and encoding/decoding added. Changed to the compound format containing both the structure and the tables.
3.2, 3.3, 3.4, 3.5, 3.6, 4.0	4	Store some additional analyzer info and also restructured the content a little.
4.1, 4.2, 4.3, 4.4	5	Changed parse data structure.
4.5	6	Changed parse data structure.
4.5.4, 4.6	7	Support for groups for constraints.
4.6.1, 5.0, 5.1, 6.0, 6.1, 6.2, 6.3	8	Changed parse data structure.

The Database Upgrade

When the TTCN Suite opens an old database, it checks if it is possible to convert the database to the current version via the TTCN Suite TTCN-MP format. A conversion program for this old database version must be available. The name of this program has a suffix which indicates the version number of the database. For example the program which is used to convert a database of version 4 is called `mp-output-old4`.

This program is installed in the TTCN Suite 6.3 release. It is used to convert databases of version 4 to MP format. The resulting MP file is then converted to GR format as usual. This mechanism solves the problem of backward compatibility. The TTCN Suite or a newer TTCN Suite version can open an old version of the TTCN Suite database.

Forward Compatibility

A special problem occurs when an old TTCN Suite version needs to open a new version of the TTCN Suite database (forward compatibility). This problem is not relevant between database versions 0, 1 and 2 since the transfer format (MP) is the same. But the MP format which is used in 3.1 and later releases have extensions. Another problem occurs because the 3.1 and later databases are stored in a single file (compound format) with extension `.itex` instead of the `.itex/.itex-tables` combo of previous versions.

In the TTCN Suite 6.3, the database contains documents of one of these types:

- Test suite
- Modular test suite
- Module

Conversion of a new database of type Module to an old 3.0x or earlier database is not possible. The only forward compatibility which is supported is for the databases of type Test Suite or Modular Test Suite.

This conversion mechanism is possible to retrofit into an older existing 3.X installation by copying the programs `mp-output-old6` and `convert-to-old6` from the 6.3 installation's binary directory for the architecture(s) installed and put these copied files into the corresponding directory in the old 3.X installation tree (these files are present in the current release for primarily this reason).

To make a 3.0x version able to convert from 6.3 databases it also needs the companion file with the extension `.itex-tables` to exist beside the 3.6 database file. Fortunately this file may be a dummy file containing nothing at all.

Known Issues

SDL and TTCN Suite common tools known issues

Supported architectures

The delivered executables and object libraries for Linux and Sun Sparc are compiled in 32-bit mode.

LP64 Architectures

Though not supported, limited testing of the generated code is performed for the SDL Suite under LP64 architectures (primarily 64 bit Solaris). For Solaris, a special version of post.o is available for 64-bit platforms (lib/sunos5lib/post64.o). The TTCN Suite has not been tested under LP64 architectures.

General

Not all tools are able to handle file names with spaces, such tools are Simulator/Explorer, CIF tools and ASN.1 to SDL tool.

ASN.1 Utilities

Undefined components of OBJECT IDENTIFIER are not reported.

Predefined OBJECT IDENTIFIER components have no value in SDL.

```
When we use PermittedAlphabet constraint for some
string type, and then assign Tuple or Quadruple,
which is out of constraint, no error message is
reported, for example, T ::= IA5String ( FROM ( "a"
| "b" ) ) t2 T ::= { 1,1 } does not produce any
messages.
```

Values for integer types with constraints outside the range of 32 bit integers, cannot be handled correctly.

ASN.1 BER/PER Encoding/Decoding

Empty SEQUENCE has no values in SDL. Consider the following example:

```
T1 ::= SEQUENCE {}
v11 T1 ::= {}
```

```
this is mapped to the following SDL:

newtype T1
endnewtype;
synonym v11 T1 = (. .);
```

The SDL type T1 do not have any values and the synonym will not be accepted by the SDL Analyzer. When mapping WITH COMPONENT constraint to SDL inline types are created on fly, information about them is not stored in the internal ASN.1 representation and type structures for those inline types are not generated for encoders/decoders generation request. This will produce compile errors for C code generated from the SDL package when Encode and Decode operators are generated for SDL types, because they will reference type information that does not exist.

ASN.1

```
  T1 ::= SET OF INTEGER
  T2 ::= T1 ( WITH COMPONENT( 0 .. 5 ) )
```

generated SDL

```
newtype T1
  String (Integer, emptystring)
endnewtype;

syntype T2 = T2_INLINE_0 endsyntype;

newtype T2_INLINE_0 /*#SYNT*/
  String (T2_INLINE_1, emptystring)
endnewtype;

syntype T2_INLINE_1 = Integer constants 0 : 5
endsyntype;
```

Information about T2_INLINE_0 and T2_INLINE_1 is generated only to SDL and is not generated to type structures for coders.

Simple workaround is not to use WITH COMPONENT, but create a new type which will be used in SET OF type:

```
  ConstrainedINTEGER ::= INTEGER ( 0 .. 5 )
```

```
     T1 ::= SET OF INTEGER
     T2 ::= SET OF ConstrainedINTEGER
```

TTCN generator: Automatic tagging is not mapped correctly to TTCN. When automatic tagging is applied, all implicit automatic tags are mapped as explicit to TTCN, because there are no means to define automatic tags for TTCN ASN.1 types. In ASN.1 automatic tagging assume IMPLICIT type of tags, but when mapped to TTCN this information is lost, and tags are treated as EXPLICIT.

```
M
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

S ::= SEQUENCE {
        x BIT STRING,
        y INTEGER
}

END
```

Types for fields x and y should have [1] IMPLICIT and [2] IMPLICIT tags after automatic tagging.

mapped to TTCN

```
S ::= SEQUENCE {
        x [0] BIT STRING,
        y [1] INTEGER
}
```

Tags for fields x and y are EXPLICIT by default, which is not consistent with ASN.1 semantics.

This problem does not influence the encoding and decoding because tags are extracted from the generated type info for each type, which is correct, and not from the TTCN internal type description.

Java Interface

In the file postmaster.java, line 53, you must change the string "libsdtpost" to "sdtpost" before compiling. Additionally you must change the name of the library on Windows, not UNIX, from libsdtpost.dll to sdtpost.dll.

Online Help

On UNIX, when using netscape as help viewer the preference netscapecommand is never used. Instead the preference firefoxcommand is used for both help viewers netscape and firefox. Also the Set Default menu command in the preference tool is never dimmed when the help viewer is set to Firefox.

Microsoft Word Integration (Windows only)

When you add/remove a link between two endpoints belonging to documents loaded in MS Word, only one of them changes character format.

Print

All used colors are displayed as white when using the option WebPrint on UNIX.

Text Editor

When you deselect text in the Text Editor on UNIX, endpoints fail to redraw endpoint underlines under the selection.

Work-around: Use <Ctrl-L> to redraw the text after deselecting a selection covering endpoints.

SDL Suite known issues

Cadvanced / Cbasic SDL to C Compiler

Combinations of component selections sometime give C compilation errors. Assume the following types in SDL:

```
newtype s struct
 a, b integer;
endnewtype;
newtype t string(s, empty)
endnewtype;

dcl v t;
```

When now the following two level component selection is performed:

```
 v(2) ! a
```

i.e. select struct component a in the second string component a compilation error might occur, as the string component selection must be performed using a function, while struct component selection can be performed using standard C ".". The generated C code will thus look something like:

```
    Extr_t(v, 2).a
```

The problem is that not all compilers accept component selection on a function result.

A suitable work around is perform the component selection in two steps:

```
    tmp := v(2),
    x := tmp ! a;
```

Cmicro SDL to C Compiler / Cmicro Library

The SDL to c compiler generates superfluous macro calls in the parameter list for operators defined in SDL. These macro calls will cause compiler errors. This only affects tight integrations.

If un-defining the flag XMK_USE_RECEIVER_PID_IN_SIGNAL it is not possible to use xRouteSignal() function in any case.

Copy & Paste in graphical editors

1007394 (Windows only): Problem with copy & paste of strings which includes Japanese characters. This problem only occurs when copying strings between two symbols.

Work-around: After copying the string, paste the string in a text editor and then copy it from the text editor and paste it in the second symbol.

CPP2SDL

If, on Unix, the CC or cc compiler is used for preprocessing the input to CPP2SDL, and messages are output during preprocessing, these messages will be present in the preprocessed file. This could lead to problems when this file is translated to SDL.

Work-around: Make sure that no messages will be issued during preprocessing by first preprocessing the files from the command-line.

Tagged types, where the name of the tag is the same as another declaration in the same scope, are not handled in an acceptable way. There will be errors when the generated code is compiled.

Example:

```
struct S {
};
int S;
```

Work-around: Change the name of either the tag or the other declaration.

Overloading on the dereferencing operator (*) is not correctly handled by the Code Generator.

Complex combinations of pointers and multi-dimensional arrays are not correctly handled by the Code Generator.

The Simulator will not be able to simulate an SDL system that uses a C++ class with public members that are inherited from a base class with virtual inheritance. However, application generation for such a system works.

In Windows, CPP2SDL sometimes reports syntax errors in the preprocessed file rather than in the header file. Such an error message is therefore not navigable from the Organizer Log.

MSC Editor

- Defect 10149: If you use the "Edit | Make Space..." command with a value larger 100 or larger the symbols below your selected object will "disappear". The error occurs, since only one page is added to the diagram. You can reclaim your symbols by increase the diagram size with the "Edit | Drawing Size..." command.
- Defect 13737: If you are running the MSC Editor Standalone (i.e. not started from the Organizer), the preference changes you make in the Organizer takes no effect automatically. In order for the changes to take effect you have to copy the sdt.ini (on Windows) or .sdtpref (on UNIX) to the directory from where you are starting the editor in order for the preferences to take effect. E.g. if you are want to try out the example in the msctrace directory, you should copy the preference file to the <install dir>/examples/msctrace/<platform>.

- When opening an MSC PR file in the MSC Editor all the notes (/* ... */) written in PR will be lost when the diagram is converted to MSC GR.
- Sometimes lines or symbols might get unconnected. If you experience this problem in some rare case, you should proceed as follows to avoid a crash: Try to drag the symbol or the (end of) the line so that it gets reconnected. If you succeed with this everything is fine again. What you definitely should avoid is to delete the symbol that the "unconnected" symbol or line was connected to before getting "unconnected", since if you then move the "unconnected" symbol or line, the editor will crash.
- When generating Encapsulated Postscript files (EPS) for multiple page Message Sequence Charts, only one file is generated containing the last page of the MSC.

SC to SDL converter

Converting SC to SDL does not work unless case sensitivity in SDL is set to OFF.

SDL Editor

Text outside drawing area:

If you type a lot of text in an object or if you open a diagram and the font that is used is larger than when the diagram was last saved, the text may flow outside the drawing area. This is visible if the text appears to the right or below the bottom of the drawing area, and the whole drawing area is visible in the window.

Connect to Text Editor:

When you select Connect to Text Editor - to edit object text in another text editor - and then exit the Organizer before you exit the Text Editor, a number of error messages may be displayed.

Work-around: Close the Text Editor or the SDL Editor before you close the Organizer.

The preference Editor*GateSymbolColor is ignored by the SDL Editor.

SDL Index Viewer

The normal search function in the Index Viewer can in some situations not find all occurrences of a text string.

Work-around: Make sure that no symbol is selected before the search begins or use the fast search function instead (by just typing on the keyboard when the Index Viewer window is in focus).

SDL Info Server

The SDL Info Server is used by the Signal Dictionary in the SDL Editor and the Type Viewer.

When using the Signal Dictionary, the SDL Info Server is not able to find signals sent in algorithms in task-symbols. E.g.:

```
task {if (IsCorrect)
  output ok;};
```

Here the signal ok will not be found.

The following examples will not the SDL Info Server be able to parse:

```
newtype a inherits b
operators
putbits: W -> W;
endnewtype ;
```

The following is OK:

```
newtype a inherits b adding
operators
putbits: W -> W;
endnewtype ;
```

SDL Simulator

Transitions in global procedures are not handled in coverage table.

Undo does not work correctly if the system contains external synonyms.

Undo does not work correctly if Define-Continue-Mode is set to ‘On’.

SDL SimUI / ExpUI

When you simulate or validate in the UI and regenerate the running simulator/explorer in the Organizer and then choose select Restart in the UI, the UI might hang.

Quotation marks that are not within a string are consumed by the UI. So e.g. patterns in SDL Explorer commands can only be enclosed in apostrophes.

SDL Explorer

Transitions in global procedures are not handled in coverage table.

Time syntype in SDL system leads to compilation error for xEq_SDL_Time

Work-around: Do not use syntype of time.

Faulty behavior if queues are added to channels to or within block instance sets with the Define-Queue command.

ANY in an expression is deterministic. Only one result is given.

Work-around: If possible, use ANY in a decision instead. The SDL Explorer will then explore all the decision alternatives.

The MSC trace of input none in the Simulator is output of none with no input. This type of MSC cannot be verified in the SDL Explorer. In the SDL Explorer there is no trace of input none.

Work-around: Remove them manually.

MSC verification of block instance sets is not correct. The SDL Explorer cannot differ between signals sent from environment to one specific instance of the block type.

Targeting Expert

Windows only: The Targeting Expert will not work if the SDL Suite installation is done in a directory path containing spaces.

Target directory structure ...

- ... is not updated correctly if an SDL system (.sdt) file contains several SDL systems and the order of the systems is changed (or a new system is inserted between existing systems).
- ... is not set up correctly if working in one target base directory using several different SDL system (.sdt) files.

When building a Threaded Light Integration for OSE with the coder library, a linker error may occur (efs_stderr not found at linking).

Work-around: Change the linking order in the generated makefile so that the OSE libs are linked in after the coder library.

Threaded Integration does not work with following options: MSC Trace, Handle time in seconds or Alternative signal sending.

Target Tester

If 'Allow large signal parameters' for the target library, the MSC trace is wrong (the output of the signal containing the large signal parameter is missing). In addition you'll get the following error in the text trace:

**ERR: Unknown trace message cannot be converted

**ERR: to MSCE message. Message will be ignored.

Complex signal parameters may lead to wrong values in the trace.

Complex signal parameters may lead to wrong values in the trace.

The Target Tester sometimes cannot establish contact to the Organizer, although the Organizer is running. Start the Target Tester again, it always works in the second attempt.

UNIX only: The Target Tester cannot remove some temporary files below the /tmp directory accordingly in some machine configurations. The result is that /tmp contains many files called /tmp/.sdt.cmicro.*. It is possible to either remove the files by hand (rm /tmp/.sdt.cmicro.*) or to use the reset command from the Cmicro postmaster $sdtbin/sdtmpm -r). This problem has to do with Solaris 2.5, which contains a bug when socket connections are to be finished.

Threaded Light Integration

When building an OSE threaded light integration using the TCP/IP module, an include path is missing in the Targeting Expert.

Work-around: Add -I$(OSEROOT)/sfk-solaris2/std-include in the "Include" text box in the "Compiler" tab in the Targeting Expert.

TTCN Suite known issues

General

TTCN Suite 6.1 on Linux can't open .itex file created on Windows or Solaris or TTCN Suite on Linux 6.2 and higher, and vice versa. The cause of this problem was in byte order in binary file format - Linux TTCN Suite 6.1 used incorrect byte order. So, if there are any .itex files created in Linux TTCN Suite 6.1 please convert them to MP format using TTCN suite 6.1 on Linux. Otherwise no Windows or Solaris TTCN Suite, or no Linux TTCN Suite 6.2 and higher, will be able to open such file.

You cannot open or convert test suites in the 3.1 format in Windows. This is because there is no version of ITEX 3.1 for Windows. If you have ITEX 3.1 on Solaris, you should export the file to MP format and import it in Windows.

On the Linux platform there is a limitation on which test suites can be loaded. Files in the .itex format stored by a TTCN suite version older than V4.6.2 cannot be loaded into TTCN suite on Linux. There are several workarounds, either open the file in V4.6.2 and save it in .mp format or open the file on Windows or Solaris platform and save the file there.

Directory list not updated on Solaris:

You can use a filter command in file selection dialogs to specify a new filter and update the file list. However, the directory list will not be updated. This means that if you recently created a new directory, it will not be visible.

MP file format problem:

When you import certain test suites from MP, there may be a problem with transferring the information (e.g. Description) in the overview tables to the tables in the dynamic part.

The TTCN standard allow path specifications to optionally include the suite identifier first. This has the unfortunate effect that if the test suite contains a top level group with the same name as the test suite there is, in general, no way of knowing if the first part of the paths is a group identifier or the test suite identifier.

ITEX assumes that if the first part of the path is equal to the suite identifier, it is the optional suite identifier and, when you import, strips it away. When you export to MP, it is always added to the front of all paths. That way ITEX is always able to import the MP files it exports. Note that ITEX v2.0 did not add the suite identifier at export, and therefore the problem described here may apply when you import MP files exported by ITEX v2.0.

The only remaining problem is importing test suites that have top level group identifiers that are equal to the test suite identifier without having the suite identifier prefixing all paths. You can avoid this problem by temporarily renaming the test suite to something unique before importing it into ITEX and then renaming it back when in ITEX.

On-Line Help

Viewer selection confusion on Solaris:

In the Solaris version, there are two separate ways to specify what application to use for displaying on-line help. You can either select it in the ITEX resource file or in the Preference Manager. If these are set differently, what application will be used depends on from where you invoke the on-line help first.

Table Editor

Handling of the tab character:

You cannot type a tab character in the Table Editor. However, a tab character from an MP file will appear, but the width will be different in the Table Editor compared to the printout.

TTCN Analyzer

Defect 4980: The analyzer will make an incomplete analysis when there is an empty behavior line following a (possibly erroneous) send or receive statement.

```
Type lookup:
```

Due to how the type lookup mechanism is implemented, the context where a type is referenced have to be seen before encountering the value to look up. This has the effect that constructs like:

```
[{a 2} = variable_with_sequence_as_type]
```

is going to fail looking up the type of the sequence value.

Work-around: Rewrite that construct to reference the variable first:

```
[variable_with_sequence_type = {a 2}]
```

ITEX does not analyze the "Type Encoding" field in "Simple Type Definitions" and the "Encoding Variation" field in "Structured Type Definition". These fields do not have any semantics in the current implementation of ITEX.

TTCN to C Compiler

Defect 6238: TTCN encoding definitions are not supported by the CCG. This is not considered an error, but rather a limitation.

3002315: If a Test Case Variable has the name Foo, and a Test Case is named FooD, this will cause a compilation error. So, there is a limitation that a Test Case may not have the same name as a Test Case Variable, plus an additional ’D’.

SDL and TTCN Integrated Simulator

Defect 6556: The TTCN Simulation kernel cannot encode OBJECT IDENTIFIER:s.

Defect 9122: On Windows NT platforms, manually sent signals to the TTCN Simulator at Simulator startup are not consumed properly.

Defect 10764: CANCEL statements without parameters does not work in the SDL and TTCN Integrated Simulation with concurrent TTCN.

Aborting an TTCN Simulator run:

Aborting the execution of the TTCN Simulator when the SDL Simulator is running, will inhibit further communication. Always restart the SDL Simulator when you abort the execution of the TTCN Simulator.

GCI / EGCI API

1008933: GciCreate function is not working and there is no plan to implement the function.

Instead, task (taskcreate, taskdelete, etc.) is recommended to use.

Chapter 1 Compatibility Notes

Chapter

2

Release Notes

This chapter contains information related to the SDL Suite and TTCN Suite 6.3 releases, in terms of:

- **Included components**
- **New and changed functionality**

Included Components

The SDL Suite 6.3 and TTCN Suite 6.3 release features the following tools and documents:

Common SDL Suite and TTCN Suite Tools and Features

- Organizer
- Print utility
- Preference Manager
- Postmaster Libraries
- ASN.1 Utilities

The SDL Suite Tools and Features

- SDL Editor
- CIF Converters
- ispell integration
- Emacs integration **(UNIX only)**
- MS Word Integration **(Windows only)**
- Link Manager
- Type Viewer
- SDL Coverage Viewer
- Index Viewer
- SDL Analyzer
- SDL Access (not included in the standard distribution)
- CPP2SDL Utility
- SDL to C Compilers (Cbasic/Cadvanced/Cmicro/Cextreme)
- Targeting Expert
- SDL C Compiler Driver (SCCD)
- Simulation Library and GUI
- Validation Library and GUI
- Application Library
- Master Library
- Cmicro Package
- Threaded Integration
- ASN.1 BER/PER Encoding/Decoding
- OM Editor
- SC Editor
- MSC Editor
- HMSC Editor

- Deployment Editor
- Text Editor
- UML2SDL Utility
- OM Access

The TTCN Suite Tools and Features

- Browser
- Finder **(Windows/Linux only)**
- Table Editor
- TTCN Link
- Autolink
- Data Dictionary
- Analyzer
- TTCN to C compiler
- Simulation Library and GUI
- TTCN Access
- TTCN Exerciser
- ASN.1 BER/PER Encoding/Decoding
- TCP/IP Module for adaptation

User Documentation

- Installation Guide – on-line help and PDF files on the installation CD
- Release Guide – on-line help and PDF files on the installation CD
- Getting Started (tutorials) – on-line help and PDF files on the installation CD
- Methodology Guidelines – on-line help and PDF files on the installation CD
- User’s Manual – on-line help and PDF files on the installation CD

New and Changed Functionality

This section contains a list of new and changed functionality in SDL Suite and TTCN Suite 6.3. The issues documented here are added or modified since the SDL Suite and TTCN Suite 6.2 (6.2.0) release.

Common SDL Suite and TTCN Suite Tools Features

- With this version the SDL and TTCN Suite now have been tested with the Visual Studio 2008 (incl. Express version) compiler environment.
- With this version the tool has been renamed to IBM Rational SDL Suite and IBM Rational TTCN Suite.
  The simulation tool previously known as the TTCN SDL Co Simulator has been renamed to SDL and TTCN Integrated Simulator.
- For information about what have been solved since the previous release, please view "Latest news" from the Organizer Help menu.