
IBM Rational Tau and DOORS Analyst 4.3 Known Problems

This documents lists the known problems with Tau and DOORS Analyst 4.3. Where applicable, remedies and workarounds are provided.

Copyrights

This edition applies to IBM Rational Tau version 4.3 and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2000, 2009.

U.S. Government Users Restricted Rights - Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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:

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.

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

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.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

UML Modeling

Workspace, Projects and Add-ins

Project related issues

Some issues related to projects are described in “Repeated add of files to project causes TAU to “hang”” on page 53.

Not possible to start Tau on UNIX

Symptom

If you open and close Tau very many times on UNIX, it will no longer be possible to start Tau.

Remedy

Restart the X server.

Tau does not work if SELinux is enabled

Symptom

Tau cannot be launched on Linux if SELinux feature is enabled. The following error message is printed:

```
<tau inst>/bin/vcs.exe: error while loading shared libraries:
```

```
<tau inst>/bin/libstudiogui.so: cannot restore segment prot after reloc:  
Permission denied.
```

Remedy

The SELinux can be temporary disabled, however the corresponding command ("setenforce 0") should be executed by root.

Active Modeler may fail to execute

Symptom

Right now ActiveModeller supports only classes and packages as root elements. If something else is created on top level, ActiveModeller will not work properly and tcl errors may appear.

Remedy

Select a class or package as root element only.

Build-in webbrowser is IE 5 on UNIX

Symptom

Tau uses Internet Explorer 5 as the build-in webbrowser on UNIX. This has the consequence that defects available in Internet Explorer 5 will also be available in the built-in webbrowser on UNIX.

Remedy

Use external webbrowser on UNIX if needed.

Platform

UNIX only.

Welcome page generates javascript errors

Symptom

If Tau/Modeller 2.7.1.1 and Tau 3.0 is installed on the same machine and one of them is uninstalled, the welcome page for the remaining tool will break.

Remedy

Run the command /regserver once for vcs.exe, for example:

```
C:\Telelogic\TAU_3.0\bin\VCS.EXE /regserver  
and the welcome page will work again.
```

Add-ins not properly loaded in multi-project workspaces

Symptom

Assuming a workspace containing several projects, and the following scenario. First loading an add-in into one project, then loading the same add-in into any other project does not enable the features and profiles as expected, although the add-in appears to be checked in the “Customize” dialog for that project.

In particular add-ins using that are loading UML profiles through TCL are affected by this limitation.

Remedy

Closing and reopening the workspace at appropriate times while adding and customizing projects may workaround the problem and allow to have multiple customized projects in a workspace.

Customized menus are not removed

Symptom

Customized menus that have been added to a project are not removed when changing active project to another project where these menus should not be present. Note: using these “unwanted” menus may lead to undefined behavior.

Remedy

Uncheck the add-in, close and re-open the workspace.

Graphical framework

SVG cannot show EMF images

Symptom

When generating SVG, EMF icons will not be shown.

Remedy

Do not use EMF.

Limitation in clipboard resulting in question marks

Symptom

The Windows EMF to WMF converter does not support unicode but depend on the Windows locale. This is due to a limitation in Windows clipboard. The result is that you might get question marks instead of the original text when pasting Tau information into other applications, for example MS paint.

Remedy

Tau manages to produce WMF with Korean/Chinese characters (or other characters if you perform the corresponding steps below) if all of the settings below are set:

1. Control panel->Regional and Language Options->Advanced->Language for non-Unicode programs is set to Korean/Chinese.
2. Code page conversion tables at the same place as above should include Korean/Chinese code pages.
3. A font with Korean/Chinese characters must be used in the diagrams. Tau uses font substitution so it will always display characters even if the characters are not available. Default font for diagrams can be set in Tau at the following location:

Tools->Options...->Format

- Developer diagram symbol font (for normal symbols)
- Developer diagram code font (for fixed text symbols)

'Developer diagram symbol font' property is used to control the font on new diagrams and 'Developer diagram code font' is used to control the font on each new fixed text symbol such as 'Text symbol', 'Task symbol', 'Comment symbol'.

Please note that the font options are only used for newly created diagrams, old diagrams have to be changed with the diagram element properties toolbar.

User zoom settings can not be read in older versions

Symptom

Loading a model created in a previous version of TAU, will automatically save the diagram settings in the new version format. If the model later is opened in the previous version of the tool, the diagram settings cannot be loaded.

Remedy

Make sure to take a backup of the .u2x files before opening the model in a newer version of the tool, if you do not intend to migrate to that version.

Tab labels disappear from the output window

Symptom

When a docked window left of the Output window resizes automatically, the Output window becomes very narrow. When the size of the Output window is restored, the tab labels are then hidden.

Remedy

Move the slider bar, that is located on the left-side of the scroll bar to the right. **Some webpages will not work on unix**

Symptom

On Linux and Solaris, some webpages cannot be shown using the Tau built-in web browser.

Remedy

For such webpages, please use another web browser.

Options and Settings

Advanced options missing on UNIX

Symptom

On Windows the “Advanced Options” has an “Operation” tab that has a check-box for “Check in even if identical”. The equivalent on UNIX is missing.

Remedy

Under UNIX there is no “Advanced Options” because this is emulated using TCL.

Platform

UNIX only.

Not possible to customize all toolbars with the Customize dialog

Symptom

The user cannot customize (i.e. remove or add buttons) all of the toolbars. Notably, attempting to add a given button is possible in one toolbar, while it is not possible to add the same button to another toolbar.

Remedy

N/A.

Diagram Generation

Editors – General

Automatic name completion

Symptom

The feature is operational in most situations but several issues are known.

- On junction and connector symbol in activity diagram: does not return available junctions and connectors

Remedy

N/A.

No automatic update of text extension symbol text in some cases

Symptom

There is no automatic update of the text in the text extension symbols if the action symbol text is a comment (`//comment`) or syntactically incorrect.

Remedy

When symbol is moved the text is updated.

Not possible to draw dependencies between some symbols and requirements

Symptom

In diagrams it is not possible to draw dependencies between a requirement and certain diagram symbols.

Remedy

Create the dependency in the Model View under the model element that corresponds to the diagram symbol, and use the Properties Editor to specify the Supplier (the requirement class). This will not make the dependency visible in the diagram, but you will get the dependency.

Solaris native keyboard cut/copy/paste keys are non-functional

Symptom

Solaris native keyboard Cut/Copy/Paste keys are non-functional.

Remedy

Use either the context menu (activated with a right mouse click), the menu bar Edit menu or the <CTRL + C>, <CTRL + X>, <CTRL + V> key accelerators instead.

Platform

This limitation impacts Solaris platforms only.

Text in text symbols representing state machines, collaborations or interactions cannot be changed

Symptom

If an entity that contains a state machine, collaboration or interaction is dragged from the workspace window to a text symbol, the symbol will correctly show the entity in textual syntax. However, the text cannot be changed.

Remedy

Edit these entities using the dedicated graphical editors instead. Visualizing the entities in a text symbol can still be useful, although editing is not allowed.

Properties Editor

Unable to see the deletion of text in the properties editor

Symptom

In some cases after entering text in the text fields, if correction is needed, then usage of delete or backspace keys to delete characters do not work as expected. This is a refresh problem.

Remedy

Deselect the text field and then select it again.

Tagged values defined using positional syntax are not supported

Symptom

If tagged values are defined using the positional syntax (not using explicit assignments to the stereotype attributes, but relying on their position in the stereotype), and these tagged values are then edited in the “Control View”, in some circumstances a value will get associated with the wrong attribute.

Remedy

Edit the values textually in the “Text View”. Avoid to use the positional syntax for defining tagged values. Use the assignment syntax.

Text Handling

Text is not displayed properly in diagrams or printouts

Symptoms

If a lot of text is present in a text symbol, the text may not be displayed in its whole. Enlarging the text symbol does not help to display the missing text.

It may also be the case that some fonts are not displayed correctly at zoom levels other than 100%.

The problems described above may happen when a diagram is displayed on screen or generated to an Enhanced Meta File (EMF).

Remedy

Adjust the zoom level to adequate value to display text on screen.

EMF generation should avoided in zoomed diagrams. It can lead to impurities in the generated file and, since EMF is a vector based format, the client program should rescale the image.

Quick undo after typing text disables redo

Symptom

Undoing rapidly after typing, while still editing text, may not introduce the appropriate 'redo step'. Redo may not work for the last few characters typed. Undo will work, however.

Remedy

Wait for a fragment of a second after typing in text before performing undo.

Shift + down arrow does not select last line

Symptom

The normal <Shift + down arrow> mode of selecting blocks of text does not work for the last line of text if it is not terminated with a new line.

Remedy

Instead, use <Shift + End> or <Shift + right arrow> for the last line.

'Select all text' command does not produce visible feedback

Symptom

When editing text, and performing a "Select All" command, the entire text is selected – but this is not always visualized on screen (i.e. the selection appears to be unchanged).

Remedy

Slightly scrolling the window to force redrawing of the text, refreshes the window contents.

Composite Structure (former Architecture) Diagram

Text on interface symbols not always shown and refresh problems

Symptom

Sometimes the text on the interface symbols disappears and cannot be edited. Also, when moving part symbols with interfaces, the diagram is not always refreshed correctly.

Remedy

Close the diagram and open it again.

Platforms

This relates to Solaris platform only.

No auto-layout option on connector lines

Symptom

In the “Tools -> Options -> UML Advanced Editing Options” for Line, auto-layout is available for all line types except the connector line.

Remedy

N/A.

State Machine Diagram

Renaming a composite state must be done with care

Symptom

The composite state machine will be deleted after being renamed, if there is a model update before the new composite state name has been typed. This happens when deleting the name first and then after a while typing the new state name. The tool will interpret it as a state deletion.

Remedy

Although composite states are visualized with an underlined name in order to highlight this potential problem to the user, in case this happens the following workarounds are available:

- Rename a composite state in the workspace window.
- Select the entire state name, and then type the new name.
- If a state machine is unintentionally lost because of renaming a state, perform an Undo to restore the state machine.

Sequence Diagram

Large diagrams makes the tool “freeze”

Symptom

When resizing a sequence diagram to larger than 10,000 millimeter the tool will freeze for a period of time (can be minutes depending on CPU capabilities) without displaying a time glass cursor. The upper limit on sequence diagrams is 1,000,000 millimeters.

Remedy

Do not resize diagrams larger than 10,000 millimeters.

Copy and paste of unconnected messages connects to lifelines in both ends

Symptom

If a message line, which is unconnected in one end, is copied and pasted it will after the paste operation be connected in both ends to lifelines. Also, if one includes other messages that are connected to lifelines in the copy and paste operation the connected messages will after the paste operation be connected to the wrong lifelines.

Remedy

Do not mix messages that are not connected in both ends to lifelines, with messages that are connected in both ends to lifelines, when doing “Copy and Paste” operations in the Sequence Diagram editor.

Model Loading and Storage **Tau on UNIX supports ASCII characters only**

Symptom

Entering a non-ASCII character in a UML model and saving the model causes an error on a subsequent load.

Remedy

To strip the non-ASCII characters from the file use the iconv utility.

This can be done as follows:

```
cp UMLModel.u2 UMLModel_orig.u2
iconv -f 8859 -t 646 UMLModel.u2 > UMLModel_new.u2
rm UMLModel.u2
mv UMLModel_new.u2 UmlModel.u2
```

Platform

This limitation impacts UNIX platforms only.

File Management and File View

Not possible to open several files at the same time

Symptom

Select the “File -> Open” menu item, a dialog appears where you can browse for files to open. However, you can only select one file at a time.

Remedy

N/A.

Text File Editing

Incorrect cursor positioning when opening text files

Symptom

Opening a text document in Tau may result in incorrect cursor positioning: instead of being placed in the beginning of the document the cursor will be placed in the end of the document. Also some symbols (e.g. single/double beginning/ending quotation marks, ASCII codes 91h-94h) may be displayed incorrectly.

Remedy

N/A.

Platform

Unix only.

Semantic Checker

Redefinition of types not supported

Symptom

In a situation where we have a subclass that inherits from a superclass it is not possible to define a type (e.g. a class or a data type) as virtual in the superclass and redefine it in the subclass.

This will show up in a model as a failure to bind attributes and operations defined in the subclass. An example:

```
class A {virtual class B {} B pb;}  
class A1: A {redefined class B {Integer x;} void Op()  
{pb.x = 1;}}
```

The identifier “x” inside “pb.x” will not be resolved.

Remedy

Add the necessary properties in the type definition in the superclass:

```
class A {class B {Integer x;} B pb;}  
class A1: A {void Op() {pb.x = 1;}}
```

TSC check missing when missing access dependency

Symptom

A TIL2369 error will be produced when building the model. The problem is common for Model Verifier, C Code Generation , Agile C code generation, Activity Simulation and Web Service Simulation

Remedy

If package P1 has <<access>> dependency to another package P2 and uses some type T from the package P2, then package P1 must have <<access>> (or <<import>>) dependencies to ALL other packages that contains types which are used in definition of the type T.

Name resolution

Cannot navigate to source of error

Symptom

For C code generation, the model successfully passes through the semantic checker, but then name resolution problems (prefixed TNR) are reported. The format of the error is exemplified below:

Subject	Severity	Messages
MyProject.ttp	Error	TNR0047: Failed to find definition of @upcastMyClassName (while looking for AnotherName). GUID...

The issue with TNR errors where Subject same as the project file (.ttp) is that double clicking such an error message in the Build tab does not result in any response from the tool. (One would expect the tool to bring you to the location in the model where the problem was identified.)

Remedy

These kind of TNR errors occur in some circumstances only, and unfortunately the source of such errors has to be looked up in the model manually. Should you encounter such error messages, please report them to Tau Support so that we can help you diagnose and remedy the cause, and also to help us improve the tool support in future versions of Tau.

<<openNamespace>> stereotype

symptom:

Name resolution can't bind dependency to a <<openNameSpace>> package when there are two matching packages with the same name, but references to elements inside <<openNamespaces>> element will bind.

For example:

```
<<openNamespaces>> package N { class A { } }  
<<openNamespaces>> package N { class B { } }
```

```
package C  
dependency to N /* this cannot be resolved */  
dependency to N::A /* this can be resolved */  
dependency to N::B /* this can be resolved too*/  
{ }
```

Remedy:

Use references to the elements inside the <<openNamespaces>> element.

Compare and merge

Using Clearcases compare with previous version

Symptom

When running Tau compare from ClearCase using the command “Compare with previous version”, Clearcase sends versions in the wrong order.

Remedy

Select both versions and compare these versions.

Using u2fileutility for compare or merge of splitted files gives unexpected result

Symptom

When the ClearCase Tree View, selecting a version and right-clicking it and use the commands Compare and Merge will call Tau graphical compare/merge (u2fileutility).

If the u2 file that is displayed in the Tree view has elements that are stored in a sub u2 file an error message that this sub u2 file could not be read appears.

Since the project is not loaded this is understandable but not expected.

Remedy

N/A.

Comparing old version of model shows differences

Symptom

If compare/merge is issued and one version of the model is in a format older than Tau 3.0 and the other version is Tau 3.0 or later, you might see differences on for example performance lines even though they are no differences.

Remedy

Save the model in the new format before the compare or merge operation is issued.

UML for Webservices

Webservice simulation

Webservice simulation is not supported on UNIX

Symptom

Webservice simulation is only available on Windows.

Remedy

N/A.

Platform

UNIX only.

UML for Model Validation

Validator UI

Validator UI gets disabled after a timeout

Symptom

Under certain circumstances the Validator UI can become disabled. This might be caused by communication failure between the Validator UI and Tau.

Remedy

N/A.

Validator connection fails if IE uses proxy server

Symptom

If Internet Explorer is configured to connect to the internet through a proxy server, an error message from the proxy is displayed when starting the Validator since the proxy may not be able to resolve the http://localhost:port/... URL correctly.

Remedy

Disable the use of proxy server in Internet Explorer

Validator UI gets disabled after a timeout

Symptom

Under certain circumstances the Validator UI can become disabled. This might be caused by communication failure between the Validator UI and Tau.

Remedy

N/A.

UML for Model Verification

Model Verifier (execution)

Sometimes “Step Into” and “Step Over” must be applied twice

Symptom

Sometimes, the “Step Into” and “Step Over” commands have no immediate effect. This happens when the next statement is transformed by the code generator into several intermediate statements, for instance in the case of a creation action (“new” keyword) of an active class.

Remedy

Apply the command again.

Not possible to step or set a breakpoint in destructors and default constructors

Symptom

The “Step Into” command does not step into destructors and implicit constructors. Breakpoints are ignored when set in destructors or default constructors.

Remedy

Use textual traces to observe the behavior of destructors and default constructors.

Not possible to open a pointer object in Watch window

Symptom

When a pointer object is displayed in a “Watch window”, it is not always possible to open the object to see its contents. It is possible to open the object only when it is the first one found in a depth-first traversal of the memory graph starting from an attribute of an active class or of an operation.

Remedy

N/A.

Edit mode cannot be entered in Watch window

Symptom

You cannot enter the edit mode (in which you can change values of variables) in a Model Verifier watch window when the window is not docked.

Remedy

Docking the watch window will allow you to enter the edit mode.

Platform

This limitation impacts UNIX platforms only.

Activity Simulation

Limitations in supported constructs

The following symbols are not supported for activity simulation:

- Accept time symbol.

Import to UML

XSD Importer

Comments not imported

Symptom

Tau does not support import of XSD comments.

Remedy

N/A.

C/C++ Importer

Old imported models might cause duplicate definitions during re-import

Symptom

For version 2.6 and earlier stereotype <<Imported>> is not used in the importer so all imported definitions has no this stereotype applied. That is why they are not deleted in Tau 3.1 and later versions. As a result duplicated definitions are found after re-import.

Remedy

Delete the existing imported definitions and do a fresh import of the header files.

C/C++ import only handles ASCII characters in file name

Symptom

It is not possible to import C/C++ files if the file or the path to the file contains non-ASCII characters.

Remedy

Change the name of the file.

C/C++ import works, but C/C++ compilation fails for C Code Generation

Symptom

The usual symptom can be described in the following general way:

- You have imported C/C++ without errors.
- Building the model works with the exception of compilation errors. I.e. the model passes through the semantic checker without problems, generation of intermediate code works reports no error, C Code Generator manages to generate C code for the model, **but** the final C/C++ compilation fails, or the linker reports errors.

Remedy

This symptom indicates that you have not properly set up the CApplication stereotype before building. Please make sure that the following is correctly set up.

- On the C/C++ import package, make sure to apply the CApplication stereotype (from the RTUtilities Add-in) and specify the include file. Note that the include file can be given either as an absolute path, as a path relative to the target directory that will be used for the build or by specifying additional include directories via the make template file of the build.
- Make sure that the same language setting (C or C++) that was specified for the C/C++ import and that is set up in the CApplication stereotype on the imported model elements is used among the build settings of your build artifact.

Refer to the user documentation / on-line help for more information about the C Application stereotype and related topics.

User defined type with the same name as UML predefined are not renamed.

Symptom

User defined types that have the same name as UML predefined datatypes are not renamed during import, and can cause error messages if they will override a reference to some predefined type. For example, if user defines CArray type and has got arrays in the model, imported model will contain semantic errors because C++ arrays are imported to UML predefined CArray type, and user defined CArray will override reference to predefined type.

Remedy

Conflicting user defined types should be renamed manually.

Platform

All.

XMI Exporter

XMI Exporter doesn't support roundtrip

Symptom

The XMI importer does not support Exporting models from Tau and importing them again into Tau. It is created to support Export to Rose.

XMI Importer

Not all UML 1.4 constructs are supported

Symptom

The XMI importer reports that UML 1.4 constructs are not supported:

- AssociationClass
- Binding
- AssociationEndRole
- AssociationRole
- Junction (Pseudostate)

The unsupported constructs are disregarded and the importer resumes with the next supported element.

Note that for import of Rose models, the Rose Importer should be used instead.

Remedy

N/A.

Parse error on UTF-8 files

Symptom

A parse error is reported although the XMI file is a legal UTF-8 file with BOM.

Remedy

The parse error occurs only on files where the encoding directive is missing. The problem is easily worked around by adding the encoding attribute `encoding = "UTF-8"` to the first line of the XML.

Example:

```
<?xml version="1.0" ?>
```

must be changed to

```
<?xml version="1.0" encoding="UTF-8" ?>
```


Actions on start transitions in state machine have no graphics

Symptom

The actions in state machine are presented in a text symbol and not as graphical symbols.

Remedy

You can use the generated text symbol as is (displaying the actions, in textual syntax) or redraw the transitions manually to add the actions.

Multiple instances of state machine exist after import

Symptom

State machine diagrams are created for state machines that are already present in the model. This will cause the generated UML model to have an undefined state with multiple definitions of the same state machine.

Remedy

Always make sure to import to an empty workspace to avoid the problem.

Text boxes in Rose appear to be unsupported by the XMI import

Symptom

Text boxes that are present in Rose models are not present in the resulting imported UML model in Tau. One would expect such text boxes to be present as UML comment symbols in Tau.

Remedy

This limitation has been identified as Rose not exporting text boxes to XMI. (Later releases of Rose XMI Export may handle these text boxes in a way such that they can be imported.)

Should the text boxes be critical to import, the only remedy is to copy and paste the contents to symbols that are supported (such as comment symbols, which are supported).

Rose sequence diagram destruction markers are lost

Symptom

After importing a sequence diagram created in Rose, lifeline destruction markers are lost.

Remedy

Such lifeline destruction markers have to be restored by editing the generated sequence diagrams.

The “abstract” property of a use case is lost after import

Symptom

A use case that is set to be **abstract** in Rose, is no longer abstract in Tau after the XMI import.

Remedy

The abstract property can be set in Tau after the import, by using “Edit Properties”.

Rose symbol color and text properties are lost

Symptom

Symbol and text properties such as font face, font size, fill color and line color that are defined in Rose are not preserved by the importer.

Remedy

N/A. Such “cosmetic” information is disregarded from when importing.

Import of nested states causes the tool to crash in some cases

Symptom

The import of nested states causes the crash of the tool in the following cases:

- The source and target of transition line are placed in the different composite states and target is History (or DeepHistory) symbol
- The source of transition line is a composite state and the target is a nested state (within the source composite state)

Remedy

N/A.

SDL Importer

Importing from Japanese edition of SDL Suite

Note

This is applicable for the Japanese edition of SDL Suite only, and can be disregarded from when importing from the standard (English) edition of SDL Suite.

Symptom

SDL systems with Japanese texts are imported, but the texts look corrupt.

Remedy

SDL Suite Japanese edition, which is supported on Solaris and Windows systems, uses “native” encoding when storing Japanese texts (such as charstrings and comments). Native in this case means SHIFT-JIS on Windows and EUC-JP on Solaris.

However, the SDL Importer requires texts to be encoded using UTF-8 and therefore files created by SDL Suite need to be converted prior to import to Tau G2. As a user convenience, the importer does this conversion transparently prior to import, by calling the `iconv` utility, a standard program for converting various encodings to UTF-8 under Solaris/Linux/Cygwin. (`iconv` is provided in the Tau installation).

The encodings that supported by `iconv` can be listed with `iconv -l`. Any of these values could be assigned to the environment variable `TauImportedSDLEncoding` in order to specify which encoding is used for the input SDL files. The useful values in this case are:

- **SHIFT-JIS** if SDL files were created using SDL Suite Japanese edition on Windows
- **EUC-JP** if SDL files were created using SDL Suite Japanese edition on Windows
- **UTF-8** if files have already been converted to UTF-8

The value of the environment variable is passed verbatim to the `iconv` utility to specify the encoding of the source SDL files. If the variable is not present or has no value, then the SDL Importer assumes that files are encoded using UTF-8.

Platform

Windows.

UML to Applications

Note that the semantics of the C/C++/Java/C# language is not identical to the semantics of the UML language. In particular when it comes down to detailed specifications using actions or expression, there can be certain incompatibilities. Some of these can be addressed by use of UML constructs provided by the profiles that represent these target languages. Others may require the use of informal (inline) code in the UML model.

Application Build – General

Closing the workspace while building may lead to loss of data

Symptom

The tool terminates unexpectedly if the workspace is closed while a build is taking place.

Remedy

Make sure to stop the build before closing the workspace. This can be done either with the “Stop Build” quick button or with the shortcut key <CTRL> + <Scroll Lock>.

Build aborted because of too long file name

Symptom

The build is aborted and the tool reports an error indicating that a file could not be created. This situation is triggered if the ASCII file name (i.e. exhaustive file name including drive and directory) exceeds 256 characters in length (a limitation imposed by the underlying operating system).

Note

When running Tau on a non-English locale, such as Japanese, Chinese, or Korean locale, the user should be aware that file names created by the build tools require more space when encoded to ASCII characters. The average expansion factor is approximately 4 times and adding a few bytes overhead. A rough estimate is a name limit not exceeding 60 CJK characters.

Remedy

Generally, in order to avoid running into this limitation, make sure that the root directory that holds the project/workspace is close to the file directory root, and avoid using longer file names than necessary.

Also, a recommendation is to use the **Build Settings** to specify the target directory of an artifact or specifying an **AnsiName** for the build root via the **Internationalization** profile. This is an efficient means in keeping the length of names of files created when building and generating code under control, and avoiding hitting the 256 byte limitation.

UML for C Code Generation

Application Build – All C Build Types

This section is applicable to the following build types:

- Model Verifier
- C Application
- AgileC Application

Partial support for active classes with internal structure

Symptom

Tau only partially supports non-single multiplicity for parts typed with active class of block type (i.e. class with internal structure, or inherited internal structure, or having children with internal structure). Having this kind of design might result in errors during code generation.

Pay attention to semantic checks during C code generation which attempt to detect such problems. For example,

```
Warning TSC0131: The multiplicity of an active class containing attribute '<name>' is ignored.
```

In general, it is recommended not to put behaviour (i.e. statemachines, operations, attributes, etc) in an active class with parts.

Remedy

There is no simple workaround for this problem. If you are going to use indexes with a part, you should not use active class with internal structure as part type. This should be taken into account when designing a model for C code generation.

Limitation with type compatibility with template instantiation

There are limitations in the C code generator when using casting operators in combination of templates. Example:

```
x = cast<CPtr<y>>(a.b);
```

The workaround is to create a temporary variable and use inline C code if needed.

Limitation when using constructors and generating code

Limitations that limit the use of constructors/destructors:

- Destructors cannot call other operations
- Default constructors cannot call other operations
- Combination of constructor with inheritance is restricted to each active class that declares the correct signature + virtual and redefined.

Destructors cannot call other operations

Symptom

Operations which are destructors cannot call other operations, except other destructors or default constructors, otherwise the generated C code does not compile.

Remedy

N/A.

Default constructor called when it should not

Symptom

When a part attribute is given an initial value, it is initialized by its default constructor (if one exists) before being assigned its initial value. This may lead to memory leaks, in case the default constructor allocates objects.

Remedy

N/A.

Remote attribute access without destination not supported

Symptom

It is not possible to access a public attribute of an active class without supplying a reference to the object that contains the attribute. This means that access using a port or only relying on the connector structure without giving any destination does not work.

Assume that we have an interface with an attribute declaration:

```
interface serv {  
    Integer i;  
}
```

The restriction means that the following will work, since an explicit destination is given:

```
{  
    Integer myi;  
    serv myserv;  
  
    ...  
  
    myi = myserv.i;  
}
```

Whereas the following will not work:

```
{  
    Integer myi;  
    serv myserv;  
  
    ...  
  
    myi = i; /* No destination given */  
    myi = i via myport; /* Access using port */  
}
```

Remedy

Use explicit destination when accessing remote attributes.

Non-const attributes cannot be defined in packages

Symptom

Having attributes in packages results in error message when generating intermediate code. This is not the case if the attributes are declared const.

Non-const attributes corresponds to having global variables in your application, without any mechanism to protect the variables (through semaphores or equivalent), which is not supported by the C Code Generator.

This is detected already when generating intermediate code.

Remedy

Such attributes must be defined in active classes, along with the code that implements the access of the attributes.

User-defined templates are not supported

Symptom

Although user-defined templates can be used when design the UML model, using such templates will cause a build error when building the application using the build types Model Verifier, C Application or AgileC Application.

Remedy

N/A.

No type compatibility of structural features with a multiplicity

Symptom

It is not possible to assign or pass as a parameter, or as a return value, an attribute or a formal parameter to an other one when they have a multiplicity, and their type is not a passive class.

Remedy

Define and use explicitly a String instance, to replace the multiple attribute.

Bad implicit de-reference for methods of templates

Symptom

The tool processes methods of templates as if their arguments or their return value were always composite. This can result in incorrect generation of intermediate format because a reference is missing.

For instance:

```
part MyPassiveClass v;  
MyPassiveClass p;  
MyPassiveClass [*] MyList;  
MyList = MyList + mkstring(v);  
// here a conversion of "v" to a reference is  
    missing in exported code
```

Symptom

N/A.

Limitations in use of packages and “output to” (using the syntax `part_name.signal_name`)

Symptom

When using the following in a state machine:

```
output <part_name>.<signal_name>  
or  
output <part_name>[<index>].<signal_name>
```

and the instantiation of the state machine containing the output statement is made in another package than the one containing the active class which is the build root, the generated C code will not compile and the compiler will output an error message like the following:

```
yPrsN_<XYZ_name_of_part>: undefined identifier  
(the exact error message depends on the actual compiler used in your  
build environment)
```

Remedy

This problem can be worked around if the instantiations with output statements are made within one package only, or by replacing the output statements that result in the compilation error with output statements using `via`, i.e.

```
<signal_name> via <port_name>
```

However, in the case of several instances of a part (i.e. the case where an index is used in the output statement), then the syntax “via followed by a port name” does not allow you to address a given instance.

C compiler reports error message with text such as “struct too large”

Symptom

The build is successful, C code is successfully generated, but the C compiler reports an error. An example how such an error will look like (in this case using the Microsoft Visual C/C++ compiler):

```
MyModel.ttp Information
c:\MyProject\MyModel_Package.h(1228) : error C2089:
'__unnamed' : 'struct' too large
```

Such errors are the result of designing UML static data structures that, when generated to C code, are too large to be handled by the compiler. A typical example where required data can become too large is arrays.

Remedy

The build tools are not aware of compiler limitations, therefore the problem must be worked around by redesigning the UML structures, or by using a compiler that supports larger structures.

Platform

This limitation is depending on the C/C++ compiler that is used, and its limitations. Refer to the compiler reference manual for more information.

Building models containing dirty .u2 files that are read-only

Symptom

When building a model, stored on .u2 files that are read-only, the tool prompts the user to confirm the save of dirty files using other file names. Answering “Yes” to that question will in some circumstances cause TRN0028 errors to be reported.

This might happen when loading a 2.3 model into TAU/Developer 2.4.

Remedy

The solution for this problem is to answer “No” and the tool will use the version of the model that is stored on the read-only .u2 file.

If that is not an acceptable solution, file attributes must be changed so that files can be written.

Explicit or implicit upcast is not supported

Symptom

This example will result in semantic errors:

```
class A { void f() {} }  
class B : A {}
```

```
A a = new B(); // Error TSC0400: When generating C code,  
implicit conversions between passive classes are not  
supported.
```

```
A a = cast<A>(new B()); // Error TSC0334: When  
generating C code, explicit conversion from 'B' to 'A'  
is not allowed.
```

Remedy

N/A

Type compatibility of template instantiations

Symptom

When building a model containing template instantiations with any of the “C” build types, the build fails with error TIL2108 and / or TIL2168.

Remedy

You have to manually define syntypes to work-around the problem. The reason for this is that the tool implementation has limitations that make it impossible to fully support template instantiations by automating the creation of required syntypes without violating language visibility rules.

As a simplified example of this, the two template instantiations:

```
P1::CPtr<int> and P2::CPtr<int>
```

are type incompatible

Defining a syntype and referring to that syntype works around the problem:.

```
syntype CPtrInt = CPtr<int>
```

“goto” in compound statements

Symptom

Using “goto” results into build failure with error message TCC0829:
Goto leading into compound statement containing attributes

Example 1

```
package p {
  active class c {
    statemachine initialize {
      start {
        {
          Integer k;
          {
            k = 1;
            {
              L1: k = k+1;
            }
          }
        }
        Integer k2;
        k2 = 1;
        {
          L2: k2 = k+1;
        }
      }
      k = k+1;
      {
        Integer k3;
        L3: k3 = k+1;
      }
    }
    Integer k4;
    {
      Integer k5;
      if (k<10) goto L1;           /* OK */
      if (k<10) goto L2;           /* NOT OK */
      if (k<10) goto L3;           /* NOT OK */
    }
  }
}
```

```
    }  
    stop;  
  }  
}
```

Remedy

Using such “goto” is considered “bad coding practice”.

This limitation has been imposed as the result of improving the support for compound statements in order to provide higher fidelity for how applications should manage attributes with respect to visibility.

Application Build – C Application Build Type

C macro expansion does not work on Windows platforms

Symptom

If “Expand macros” is turned on in the general build settings, and if the “TargetKind” in the C Code Generator settings has the value Win32, then the compilation of the build types Model Verifier and C Application will fail.

The compilation however works fine for “VxWorks C Application” targets.

Remedy

N/A.

Application Build – Compile & Link Errors

In some circumstances, the compiler will report errors even though the model is accepted by the code generator. These errors are in most case the result of a model in which limitations are not properly detected by the build chain tools. Here is a list of the most common causes of such compile or link errors:

- Destructors cannot call other operations
- Limitations in use of packages and “output to” (using the syntax `part_name.signal_name`)
- C macro expansion does not work on Windows platforms

- C/C++ import works, but errors are reported, referring to incorrect use of CPtr type
- C/C++ import works, but C/C++ compilation fails for C Code Generation
- C compiler reports error message with text such as “struct too large”

Internationalization

Charstring operators are not supported for Charstring literals containing DBCS characters

Symptom

At C code generation, all Charstring literals are translated to the user's OEM code page, which is based on the user's locale settings. For example, in the case of a Japanese locale the OEM code page is Shift-JIS. The Charstring operators from the C Code Generator do not support multi-byte characters. So Charstring manipulation operators will not work correctly on Shift-JIS characters.

Remedy

The user must implement operators which use the standard C library `mblen` function which calculates the length of a multi-byte character. These operators can then properly handle these characters.

UML and C#

C# code generator doesn't support non ASCII characters in path

Symptom

C# parser from can't parse files if the file path contains non ASCII characters.

Remedy

Do not use non ACII characters in the file path.

Roundtrip removes text diagram when operation parameter is added in C# code

Symptom

Text diagrams displaying the implementation of operations are removed if parameters are manually added in the generated code and then synchronized back into Tau.

Remedy

Do not use roundtrip in these situations or make sure the parameters are added in the UML model, not in the generated code.

<<informal>> stereotype is not supported

Symptom

The C# code generator doesn't support the <<informal>> stereotype completely. Only informal packages will be skipped during code generation; for other entities the <<informal>> stereotype has no effect when generating C# code.

Remedy

N/A.

File-level C# attributes are not updated correctly

Symptom

Certain C# file-level entities, such as 'using' declarations and attributes, are not correctly handled by the "Update Model" command (roundtrip).

Remedy

If this information is needed it is recommended to represent the file scope with a package stereotyped by <<global namespace>> stereotype, and attach the corresponding UML entities to that package

UML and C++ Breakpoints on Actions within a Decision answer

Symptom

Setting breakpoints on Actions within a Decision Answer does not result in expected results. The code that is generated for such actions is currently not instrumented.

C++ Application Generator

<<noScope>> stereotype is not supported

Symptom

Packages stereotyped by the stereotype << noScope >> are translated to ordinary packages, and references to such packages will be translated as references to ordinary packages.

Remedy

N/A.

Operations in datatypes

Symptom

Datatypes are translated to plain C++ enums. The reason for this is that operations in datatypes are not supported.

Remedy

N/A.

Signal inheritance

Symptom

Signal inheritance is not supported.

Remedy

Specifying an explicit model-to-file mapping

Symptom

When specifying an explicit model-to-file mapping, do not apply both the << manifest >> and << manifest implementation >> stereotype on the same dependency as this sometimes causes problems with missing implementations in the generated file.

Remedy

Instead use two separate dependencies.

Inheritance of state machines

Symptom

Inheritance of state machines is not supported.

Remedy

N/A.

Manually edited text in generated files may be deleted

Symptom

Manually edited text in generated files, e.g. comments and user sections, is subject to be deleted when the file is regenerated. This happens when an element is deleted from the model, and thus will be deleted from the file, and the manually edited text is located immediately before that deleted file element.

Remedy

- This problem occurs only if the flag “Support Roundtrip” is on. If roundtrip is not used this flag should be turned off.
- Add such user code before elements that will not be deleted, for example before the last `#endif` in a header file.

Manually edited text after the last generated element in a file will always be deleted

Symptom

Manually edited text after the last generated element in a file will always be deleted, when the file is regenerated.

Remedy

This problem occurs only if the flag “Support Roundtrip” is on. If roundtrip is not used this flag should be turned off.

Generating C++ for a model originally developed for C code generation

Symptom

In general it is not possible to generate C++ for a model originally developed for C code generation, without modifications. The following compatibility issues are known:

- Implicit connectors are not supported.
- Connector splitting (based on the dynamic type of the signal) is not supported.
- The “Sender”, “Offspring” and “Parent” expressions are not supported.
- The containers in the RTUtilities package are not supported. This package is intended for **C code generation only**.
- Not all operations of the predefined package are supported by the C++ code generator. See the comments on a predefined operation to know if it is supported or not.

Remedy

N/A.

Tau Object Runtime Library (TOR)

Termination of classifier behavior state machine

Symptom

If a classifier behavior state machine is terminated, the owning active class instance will not be automatically deleted, as specified in the UML semantics. Instead the instance has to be manually deleted.

Remedy

N/A.

Problems with finalizing threads

Symptom

Although the TOR supports threading, some problems with finalizing threads have been found. For example, the implicit termination of a thread when the application terminates may sometimes cause a run-time error in the generated application.

Remedy

N/A.

UML and Java

Eclipse Integration

Java support

Deactivating JavaApplication add-in deletes the <<javaSettings>> stereotype

Symptom

When deactivating the JavaApplication add-in information in the <<javaSettings>> stereotype will be deleted. Therefore, if you do this with the intention to use the Eclipse integration remember to set the correct Java language version to be used.

Remedy

Set the <<javaSetting>> stereotype manually.

Java roundtrip uses "." and "::" incorrectly

Symptom

When importing Java code on the form “x.y” there are two possible mappings to UML:

- 1) x.y (in case x is a structural feature)
- 2) x::y (otherwise)

Currently Java roundtrip doesn't know during parsing what 'x' refers to, so it has to guess (and sometimes makes a wrong guess).

Remedy

Use parentheses using the form: (quux::bar).foo().

Java5 foreach not correctly supported

Symptom

The Java5 construct foreach is not supported in Tau.

Remedy

N/A.

Java files containing Unicode escapes cannot be imported

Symptom

When importing a Java source file containing Unicode escapes an error message is issued. The contents of the file are not imported into the model at all.

A Unicode escape is a sequence of ASCII characters in the form: \uxxxx, with xxxx a hexadecimal value representing the Unicode character which encoding is xxxx.

Note: A Unicode escape is not the same as a Unicode character. Real Unicode characters can be imported since the UTF-8 encoding is supported.

Remedy

Make sure the file does not contain any Unicode escapes. Use UTF-8 encoding and Unicode characters instead.

Cannot import Java files if file name contains Korean characters

Symptom

Importing Java code stored on files that contain Korean characters results in fatal error TSX1024: cannot open input file

Remedy

Rename files and directories in the path to suitable ASCII names.

Platform

Korean locale on Windows.

Java update model limitation on comments and attributes

Symptom

When updating the model based on changes in the java code, graphical elements like comment symbols and association lines may disappear. The model will still contain the right values.

Remedy

Show the comments and the attributes in text diagrams instead.

General for COM and C++ API

Changes to the API in Tau 3.1

Symptom

The following change in the Tau COM and C++ APIs may require clients to be updated:

A new method `OnAfterVisitedEntity` has been added in the `ITtdMetaVisitCallback` interface. Client code which implements that interface must thus add an implementation of that method. The implementation can be empty.

In addition the Tau COM API now uses a version number (same as release version) on the Prog Ids. A client of the COM API should not assume that the CLSID of the COM objects will remain the same. Instead it should get the CLSID from the Prog Id (either the version dependent or version independent, depending on the needs of the client).

COM API

Building COM clients incorrectly may cause TAU to terminate unexpectedly

Symptom

If a COM client is build using mixed run-time libraries, the COM client might cause TAU to terminate unexpectedly.

Remedy

If you build a debug version of a COM client you still must use release version of the run-time libraries. This is the same concept as described in the on-line help how to debug agents build using the C++ API.

C++ API

APIError messages sometimes truncated.

Symptom

When using the C++ API with gcc 3.4 API error messages might be truncated.

Remedy

Use gcc 3.2.

Platform

This limitation impacts Linux Red Hat 4.0 only (since that is the only platform where gcc 3.4 is supported).

Common Reference

TAU-SYNERGY/CM integration

Links for merged projects and objects missing

Symptom

Merging from within TAU does not result in merge links in SYNERGY/CM.

Remedy

N/A.

Repeated add of files to project causes TAU to “hang”

Symptom

When running SYNERGY 6.4 ActiveCM patch level 2 or 3 together with TAU, performance degradation may be observed as one adds files to a TAU project. Eventually TAU might stop responding and terminate unexpectedly.

Remedy

We recommend not using these versions of SYNERGY ActiveCM together with TAU 2.6. Should you have patch 2 or 3 installed since before, you could proceed as follows to unregister the faulty DLL:

Open a DOS command window and “cd” to the “bin” directory of your SYNERGY/CM installation. Typically this would be:

```
C:\Program Files\Telelogic\SYNERGY CM 6.4\bin
```

Run the following command:

```
regsvr32 /u CMExplorer.dll
```

You may need to reboot you computer for this to take effect.

Complete task: Tau is not updated with the checked-in icon

Symptoms

In some circumstances, when a SYNERGY task is completed, the SCC status of the corresponding files (showing if the file is checked in or checked out) is not updated accordingly. In some other cases the files are checked in at a time later than anticipated, resulting in a warning that the file is read-only.

Remedy

The operation “SYNERGY->Refresh” will update the SCC status for all loaded files.

Complete task: Tau is not checking in files

Symptoms

In some situations on Linux, files are not properly checked in when completing a task. This can happen if the task is automatically created as part of a check out operation.

Remedy

Create the task manually before checking out the files.

Platform

Linux only.

Fail to detect Synergy 7.1 client installation

Symptom

If the user don't have Admin rights when installing Synergy 7.1 client, the Tau installer will fail to detect the Synergy 7.1 client installation. Note that the Tau installer still requires Admin rights to be installed.

Remedy

User with Admin right should install the Synergy 7.1 client on the machine.

TAU – SCCI

Second Tau might not start if Clearcase is started from Tau

Symptom

If Clearcase is started from inside Tau and not exited before Tau is exited, Tau will not exit when closing down Tau, it will run in the background. Starting Tau again will fail.

Remedy

Do not start Clearcase not from within Tau or make sure Clearcase is closed before closing Tau.

Adding a file to Source Control

Symptom

The underlying source control provider may impose restrictions on the files added to source control that can cause the ““Add to source Control” operation to fail. Common restrictions relate to the file location, file names and user privileges.

For instance, when adding a file with “Add to source Control”, an error message is printed. Example of such error message when the underlying CM system is ClearCase could be like:

```
Internal Error detected in ../sum_view.cxx" line 819
```

Remedy

Refer to the documentation of your source control provider for detailed information.

Tau – DOORS integration

Using multiple DOORS clients at the same time doesn't work

Symptom

If several clients and/or versions of DOORS is running on the same computer it is not possible to work with models stored in DOORS.

Remedy

Make sure that only one DOORS client is running when working with models stored in DOORS.

Not possible to use System Architect-DOORS integration together with the DOORS Integration

Symptoms

Due to a defect in the System Architect-DOORS integration, it is not possible to use the Tau-DOORS integration at the same time as the TAU-DOORS integration.

Remedy

Please contact System Architect support for a temporary resolution to this problem.

Link is not visible in TAU workspace if a baselined copy of the formal module is imported

Symptoms

Links between objects in baselined modules are not visible if they are imported into TAU.

Remedy

N/A.

Locate opens both the baselined copy and the latest version of the module

Symptoms

When locating an object in a baselined module from TAU, both the baselined version and the latest version of the module is opened.

Remedy

N/A.

Printing large diagrams

Symptom

It is not possible to print large UML diagrams properly from DOORS. Wide diagrams are not printed in their full width. High diagrams are not printed in their whole.

Remedy

The OLE size needs to be adjusted for diagrams to be properly printed. This can be easily done by using the formal module option: “Tools -> Support Tools -> Set OLE Size for Current Object”.

This option can be used to set the OLE to a size that can be printed. In the same menu there is also an option to revert the OLE back to its original size.

Platform

Windows.

Tau – System Architect integration

Certain model elements created in Tau not visible in SA

Symptom

When using the SA Integration to store Tau model elements in SA, certain model elements are not shown in the the SA Explorer (but they are saved and can be used by Tau).

Remedy

N/A.

Tau - MSVS 2008 integration

Build failed when MSVS2008 is using .NET 3.5 framework

Symptom

Build failes after creating new project in MSVS2008

Remedy

In MSVS 2008 change the Target Framework to 2.0. Tau does not support later versions of the framework

Incorrect documenation on Tau C++ API MSVS support

Symptom

Tau C++ API supports Visual Studio 2005. In the documentation it is stated that Tau C++ API supports Visual Studio 2008, this is incorrect.

Documentation and Licensing

Help and Documentation Limitations

Help functionality limitations

- In some cases, when you click help buttons in dialogs or press F1 when a dialog is open, the on-line help will not always be opened on the correct topic.
- “What’s this” help is only partially implemented.
- Screenshots of the user interface in the help may contain minor errors or discrepancies from the actual implementation.

Tau Gateway supports dongle

Symptom

The Licensing Guide document states that Tau Gateway doesn’t supports a dongle even though it does.

