Telelogic Publishing Engine

Tau Quick Reference Guide

Release 1.0

Before using this information, be sure to read the general information under Appendix, “Notices”on page 25.

This edition applies to **VERSION 1.0, Telelogic Publishing Engine** and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of contents

Telelogic Publishing Engine - Tau Quick Reference

Data

TPE currently supports DOORS, Tau and generic XML data sources.

Queries and contexts

As mentioned in the Document Template section of the reference manual, a TPE template specifies the data to be extracted using *queries*. A *query* is a path in the data source schema.

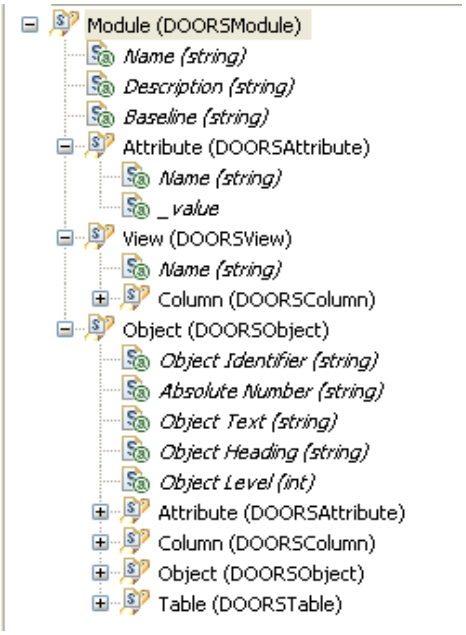


Figure 1 Sample DOORS Data Schema

NOTE For easier identification attributes are rendered in italic.

The Document Studio abstracts users from much of the complexity of manually writing queries with features such as drag and drop of schema elements. Nevertheless, it is still useful for template authors to understand the concepts of schema and queries and how they are constructed.

In the above data source schema some valid queries include:

Query	Description
module	returns a single result, the source module
module.object	Returns all the objects in the source module, as filtered/sorted by the source view
module.object.attribute	If used in a module.object context returns all the attributes for the current object.

A query can exist only in a template element. The template element and all its children can use the attributes of the entities returned by the current query and all of the queries from parent elements.

In the above example, if the query is *module.object* than any Object attribute from the schema can be used: Object Text, Object Heading etc

TPE template elements can be nested. Setting queries on elements and their children defines context. The query in the child element will be executed in the context of the parent's element query results.

Example:

Element 1: module.object

Element 1.1 (child of Element 1) : module.object.attribute

The second query will only return the list of attribute names for the current object returned by the query of element 1. In element 1 only the attributes of DOORS Objects can be used while in element 1.1 the attributes of DOORS *Object attributes* can be used (i.e the names of those Object attributes).

NOTE TPE Studio fully assists the user in building the queries and assigning the appropriate contexts. At no time you will be required to manually type a query.

Filtering data

Sometimes not all the data source is needed. In these cases you can limit the amount of processed data by setting a *filter* on the query. You can specify a filter in two ways:

- *TPE Filter* – Javascript expression using the data attributes of the entities returned by the query
- *native filter* – plain text that specific to each data source type.

When the query is performed, only the data entities matching the filter will be included in the output.

NOTE Not all data sources support native filtering.

NOTE For those data sources that support native filtering it is mandatory for the native filter to be a valid filter. TPE cannot and will not perform any validation on the native filter. Providing an invalid native filter can have results ranging from incorrect data in the output to the tool crashing.

NOTE For data sources accessed through TPE's Generic XML input driver it is not possible to define native filtering. The only exception to the rule is for the data sources where the filtering can be specified in the URL.

NOTE It is more efficient for filtering to be performed by the data source so whenever possible it is recommended to use a native filter as it should yield better document generation times than when a TPE filter is used.

Sorting data

Query results can be sorted. You can specify a filter in two ways:

- TPE Sort – the list of attributes and the sort direction (ascending/descending)
- Native sort – plain text that specific to each data source type. For DOORS this text must be in the format of the DOORS Sort

When the query will be performed, the elements will be displayed in the output document in the correct sort order.

NOTE Not all data sources support native sorting.

NOTE It is more efficient for sorting to be performed by the data source, so whenever possible it is recommended to use a native sort as it should yield better generation times than when a TPE sort is used.

Tau Data

A Tau source is defined by a single Telelogic Tau Project (*ttp* file). TPE can extract Tau data as long as a Telelogic Tau 4.2 or later software is installed on the same machine.

NOTE With the current build you need to add Tau *bin* folder to the system's PATH environment variable.

Tau Schema

The Tau schema used by TPE is automatically generated from a Tau metamodel using the provided GenerateSchema.tcl. The script is located in \s*ource**Tau*\s*chema* in the TPE installation folder.

NOTE The predefined Tau schema is built from the Tau metamodel. In order to build Tau documents you need to be familiar with the metamodel structure and its relationship with the model displayed in Tau's model browser.

The schema view will display all the elements reachable from the root element (usually model).

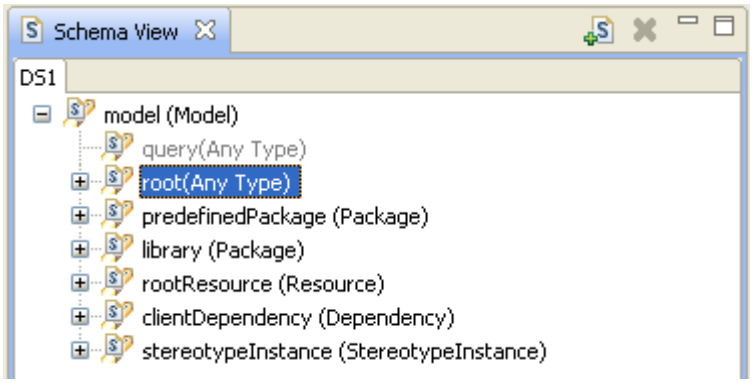


Figure 2 Tau Default Schema

Queries

A TPE query on a schema is a path in the schema, ex: *model.rootResource*. A query is applied to a TPE template element and defines the data context for that element and all its children.

NOTE The TPE syntax for the queries is similar to the XPath syntax. A major difference between the two is that a TPE query does not specify the filter in the query. TPE defines the filter and sort clauses separate from the query itself.

NOTE Each schema element (excepting the query) has an underline Tau native query (expressed in OCL) that will be used to fetch Tau data. For example, the *root* element under the *model* element has the following Tau query attached: *GetModelRoots()*

Nested queries

Adding queries to template elements and their children creates nested query.

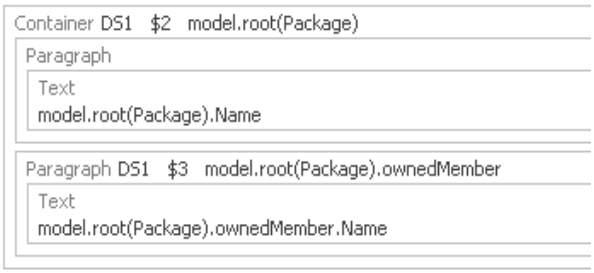


Figure 3 Nested queries

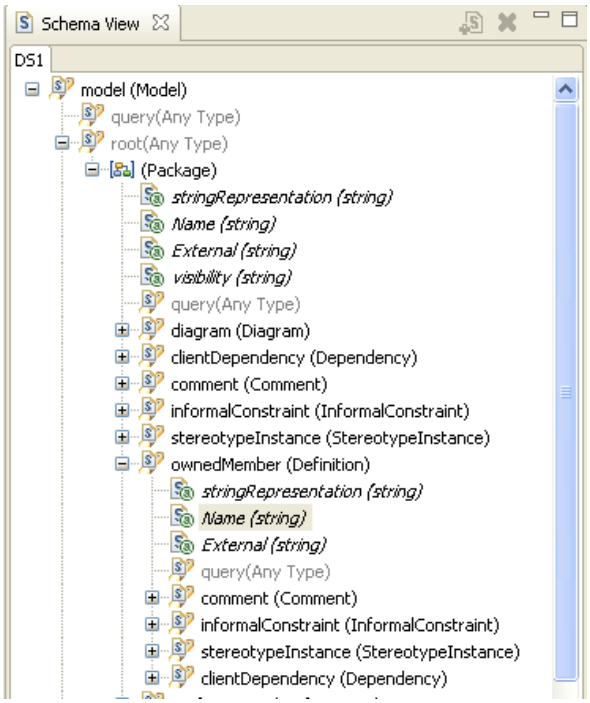


Figure 4 Schema

A nested query will be executed in the context of the results of the parent element. The first query, *model.root(Package)* will be performed in the context of the Tau model. The second query, *model.root(Package).ownedMember* will be performed in the context of each package returned by first query.

How TPE builds queries

Using the above example, if the user wants the list of all classes from the top level packages in model, the query would be: *model.root(Package).ownedMember(Class)*

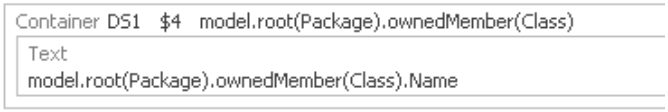


Figure 5 query

While in this form the output document will no longer contain the name of each package, TPE will build the class list in the same way as it did in the first case. The query is split into its component queries, and each query is executed in the context defined by the previous queries:

Subquery	Context	Result
model	-	the model
model.root(Package)	model	list of packages
ownedMember(Class)	list of packages	list of classes

Each sub query is performed once for every element in the context, and the results of each execution are concatenated. These results become the context for the next sub query, or the results list if the sub query is the last one.

Filter

For Tau Data Sources both native and TPE filtering is available.

Native Filter

The native filter for a query is specified in the “native filter” tab of the filter editor in Document Studio. As for any native filtering, the job is delegated to Tau, with no processing done in TPE.

A Tau native filter must be a valid Tau OCL script that returns a collection of elements. The native filter is concatenated to the underlying Tau query for the current schema element.

Example

Schema element:

model.predefinedPackage

Underlying Tau query:

GetEntities(“predefinedPackage”).select(IsKindOf(“Package”))

Native filter:

select(HasPropertyWithValue("Name", "Predefined"))

The query that will be performed in Tau is:

GetEntities(“predefinedPackage”).select(IsKindOf(“Package”)).select(HasPropertyWithValue("Name", "Predefined"))

Scripted filter

The scripted filter for a query is specified in the “script filter” tab of the filter editor in Document Studio. Scripted filtering is performed by TPE itself and follows the same rules and limitations as for any other data source.

Sort

Tau data sources do not have support for native sorting. For these data sources only TPE sorting can be used.

Type casting

Some schema elements do not have a type assigned to them as there can be more than 1 valid type for that. In such cases you can define which type to use through the “Cast to type” function in the schema view bar.

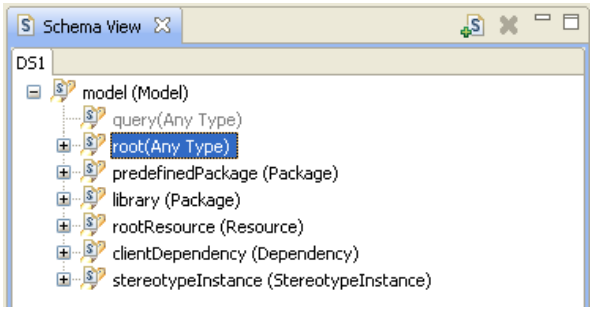


Figure 6 An element with no type

NOTE A cast query will filter the results of the regular query to return only the elements that can be casted to the selected type. Internally this is implemented by adding the *select(IsKindOf(“Type”))* filter to the query.

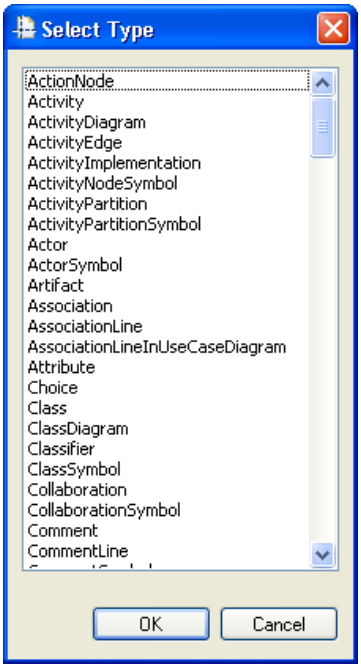


Figure 7 Select type

After the type is selected, it becomes available in the Schema View under the “anyType” element.

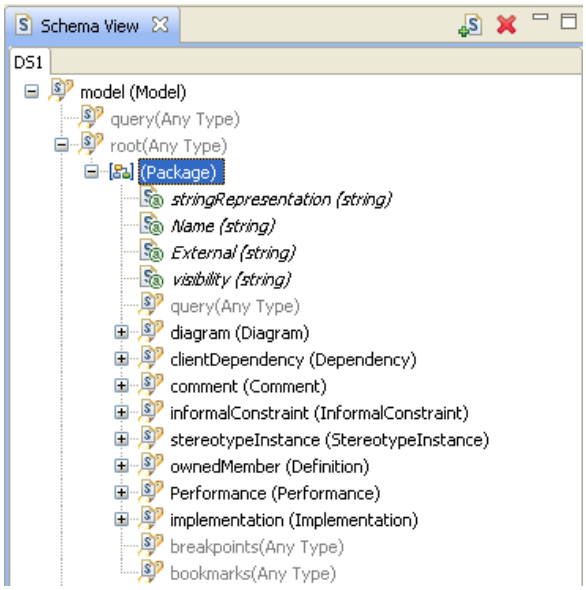


Figure 8 A "cast" to Package

Build queries using the “cast” allows access to all the child elements and attributes of the cast type..

```
Paragraph DS1   $5   model.root(Package)
  Text
  model.root(Package).Name
```

Figure 9 A query using a cast type

Type casting can be used also when you want to refine the results of a query. Basically type casting works as a secondary filter for Tau elements.

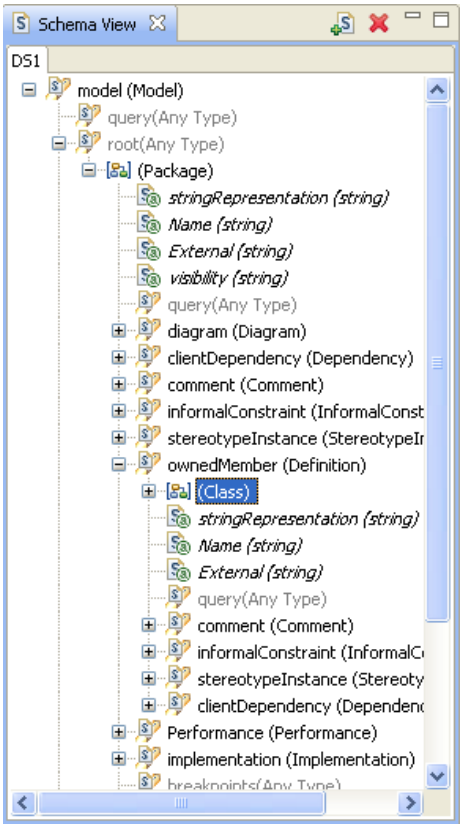


Figure 10

Adding a “Class” cast to the “ownedMember” element of a Package will allow defining the following query:

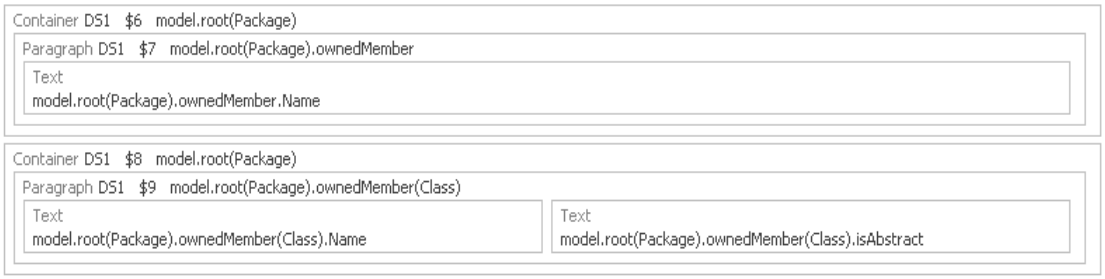


Figure 11

In the above example the query $7 retrieves all the definitions contained in each top level package while query $9 will return only the definitions that are classes from the same context.

NOTE With the current TPE version you can specify casts that are not correct (ex: from Class to Package). This queries will yield no results.

Cast vs. Filter

The result set returned by a *cast query* is identical with the result set returned by query having a filter using an equivalent *IsKindOf* predicate. The main difference is that using a cast query gives access to the cast type attributes and child elements while using a query with a filter does not.

Attributes

Once queries are assigned to template elements, all the attributes of the elements returned by the queries can be used.

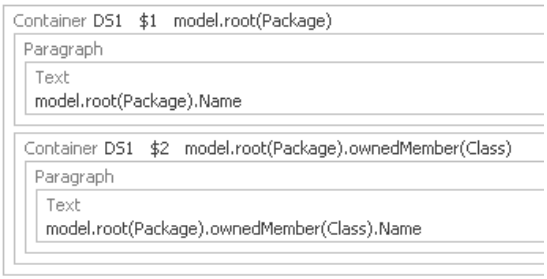


Figure 12

An attribute can be used in multiple ways:

- To define the content of a template element
- To be used in expressions calculating the content of a template element or its formatting capabilities

Special attributes

GUID

Every schema element has a special attribute named “guid”. This attribute is filled by TPE with the unique GUID of the current model element.

_image

Every schema element representing a Tau diagram has a special attribute named *_image*. Using this attribute will generate an image file for the current diagram, image that can be included in the output.

stringRepresentation

Available for all expressions, actions and definitions, this attribute holds the unparsed representation of the element.

The query element

Every schema element has a special child element named “*query*”. Unlike the other elements in the schema, a “*query*” has no underline Tau query assigned and no type. Using the query element as is will return no results.

The purpose of the query element is to offer another level of customization for the TPE document generation. Should the existing elements should not suffice (or be optimal) for a given task, the user can define what he wants to extract (the type) and how he wants that extracted (the query) using the query element. The *type* is defined by adding a *type cast* to the query element while the *query* is defined in the template element’s filter, as *native filter*.

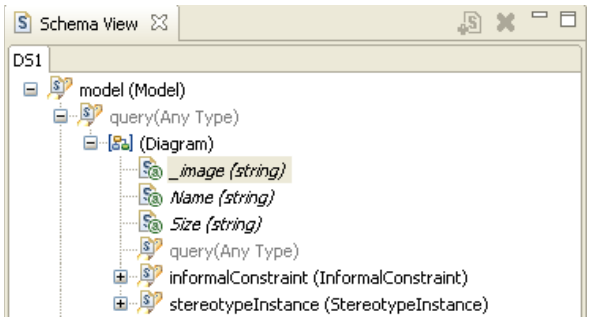


Figure 13 A cast added to a "query" element

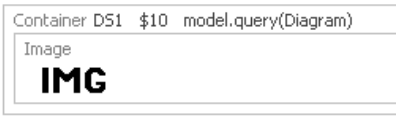


Figure 14 The query element with cast is used in the template

For the above scenario, a valid and meaningful query could be:

GetAllEntities().select(IsKindOf("Diagram"))

NOTE TPE will filter the results of the query based on the type cast used. If the query is syntactically correct, the result will contain only elements matching the specified type cast.

Tau Schema Discovery

TPE 1.0 does not offer an integrated schema discovery wizard for Tau. To generate a Tau schema for your model you need to do the following actions:

Start Tau and open your model

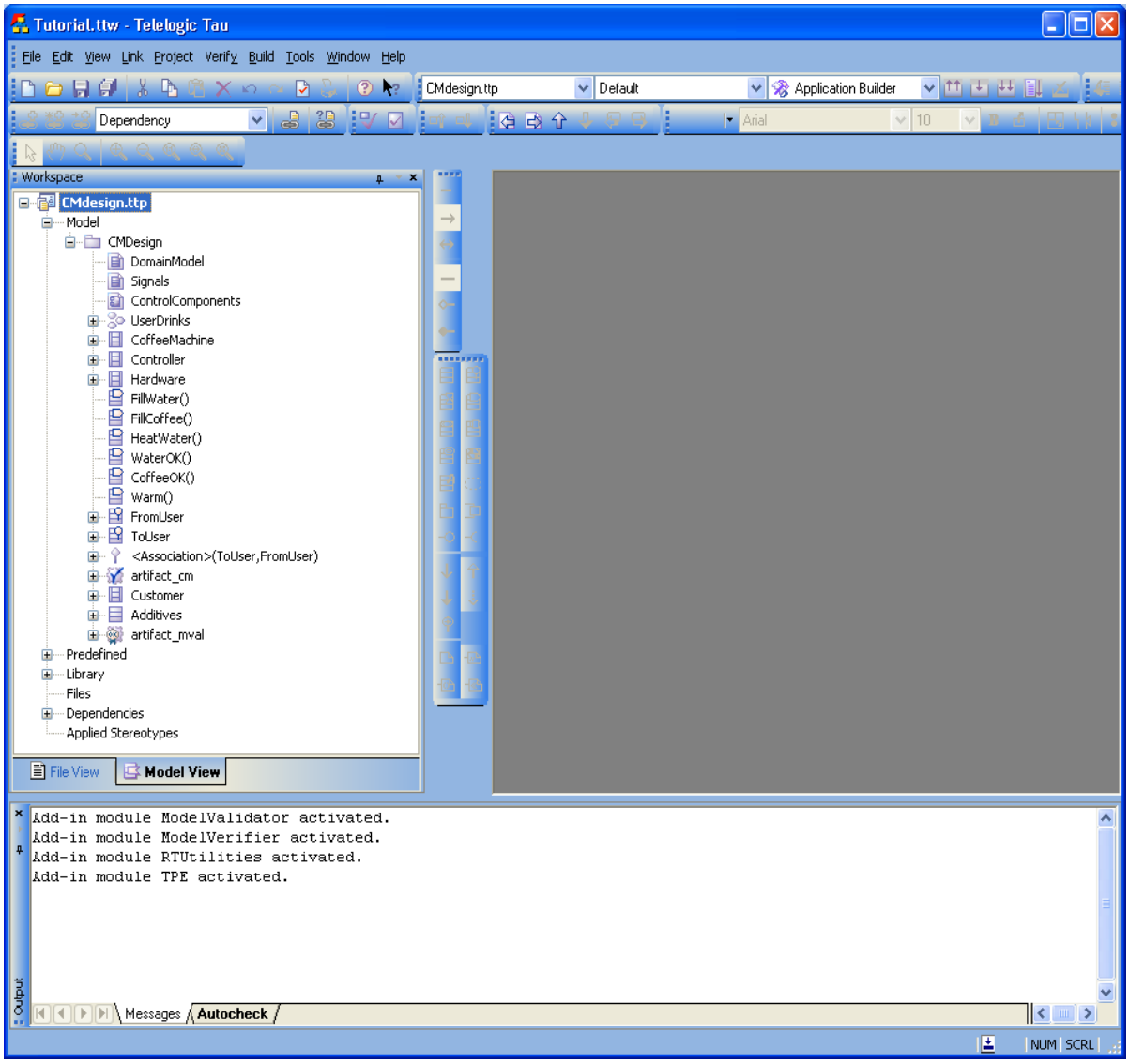
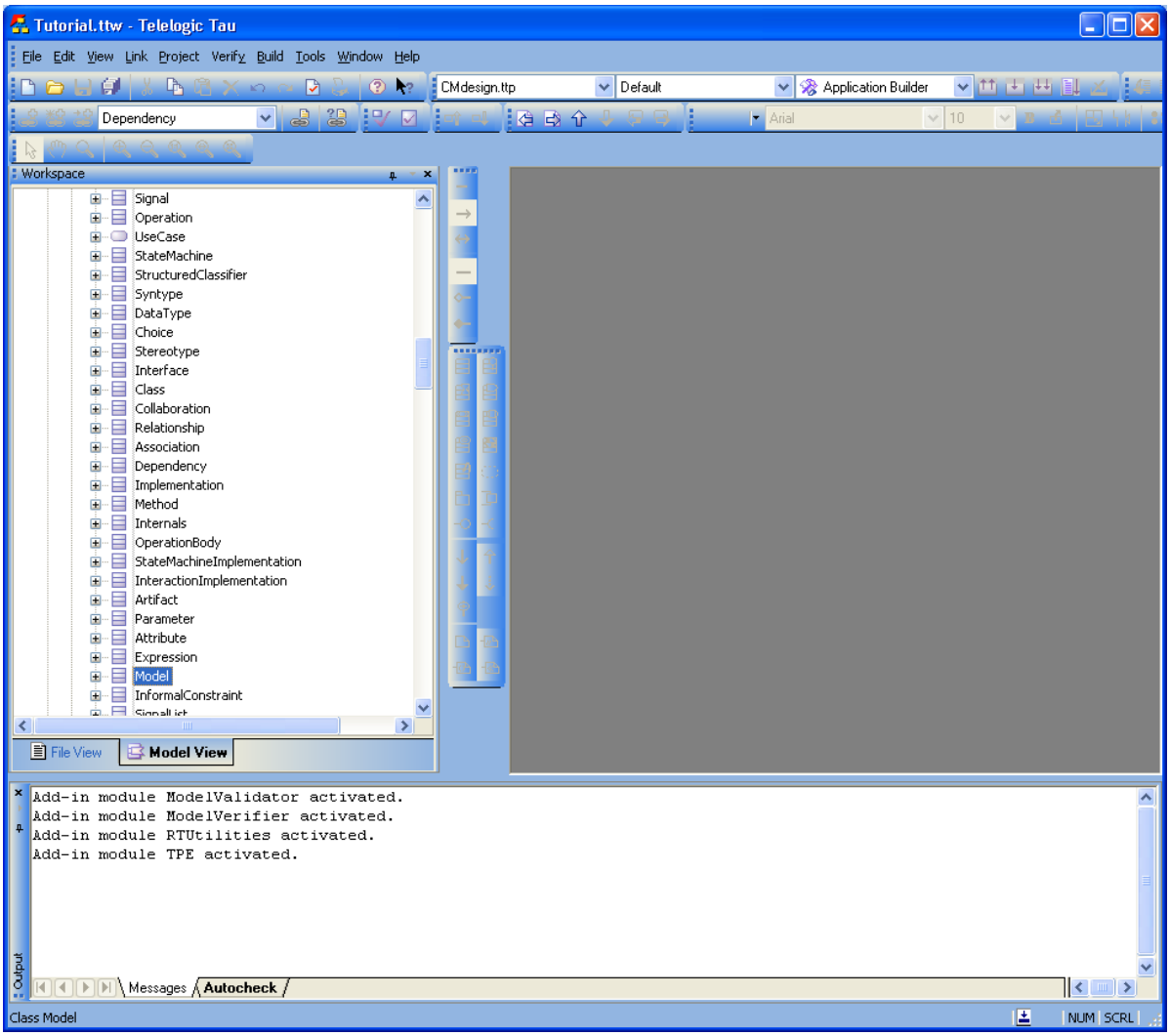


Figure 15

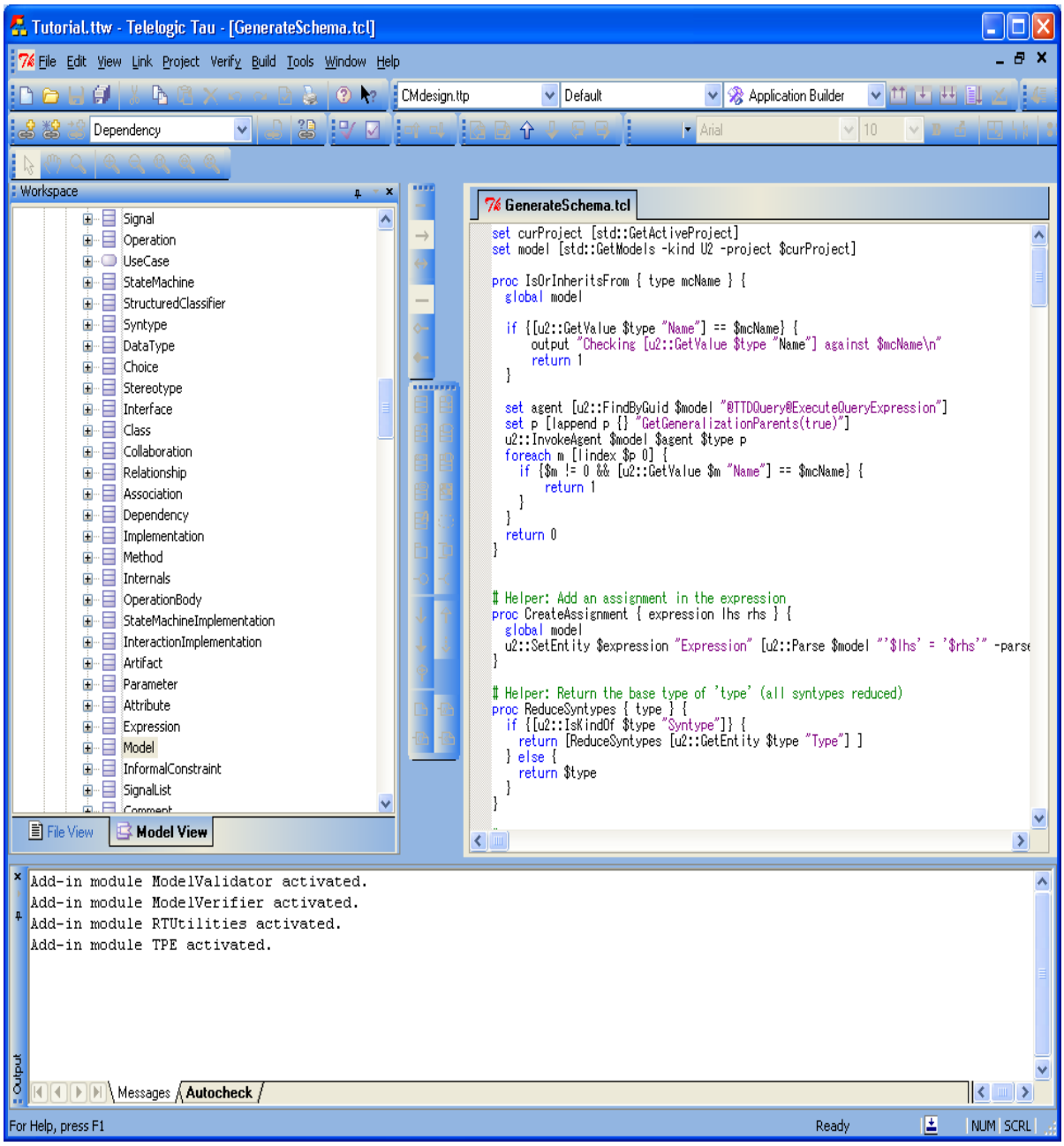
Select the metamodel's root

In the explorer select the model element that serves as your model root.

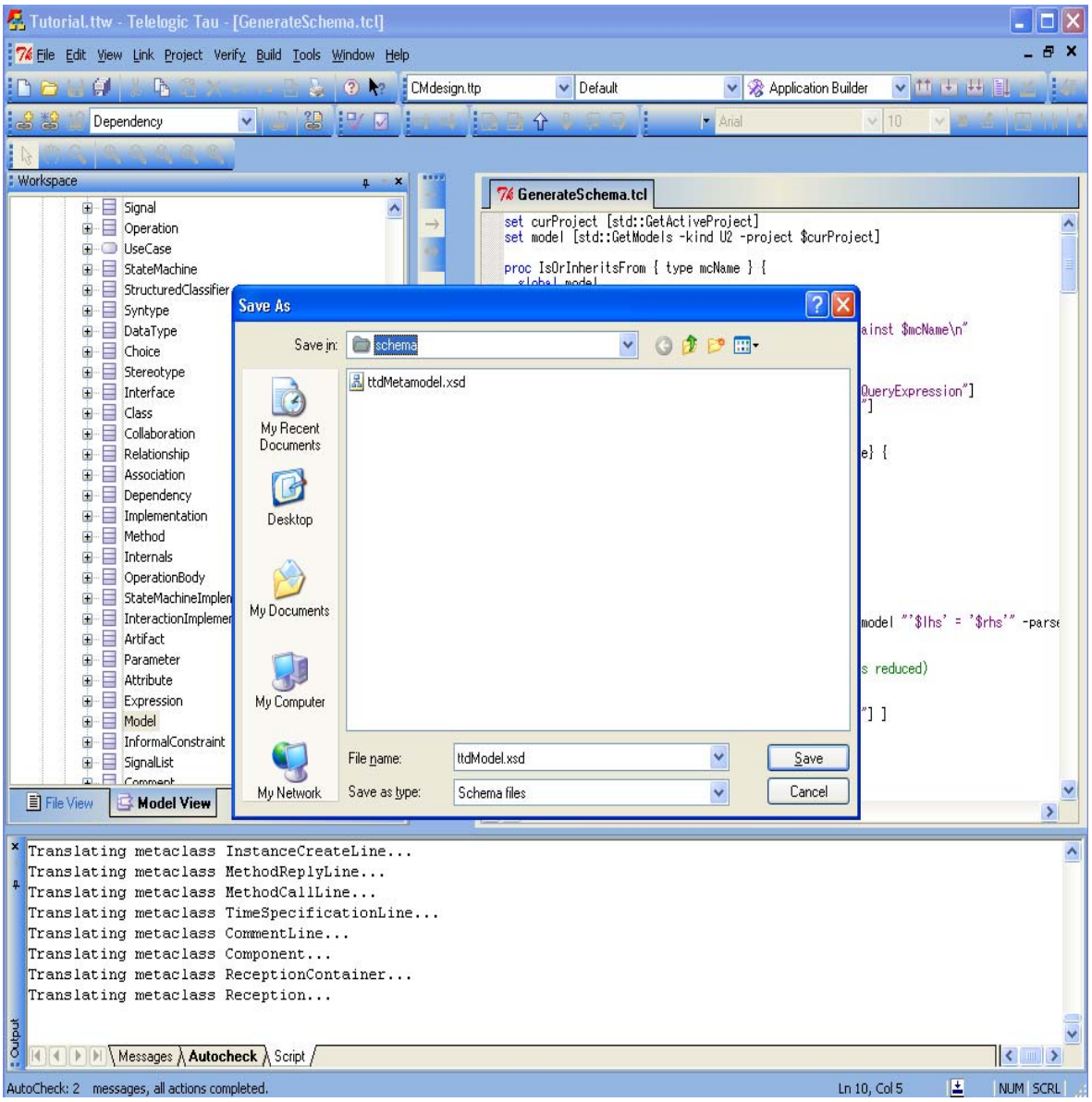


Load the GenerateSchema.tcl

Load the TCL script file provided in the TPE installation.



Run the script



Once the script execution is complete you will be prompted with the location for saving the schema

Tau Addin

Installation

The TPE addin for Tau is automatically installed by TPE if a valid Tau installation is found. If you install Tau after you have installed TPE you can install addin by running the TPE installer again with the option to modify the existing installation.

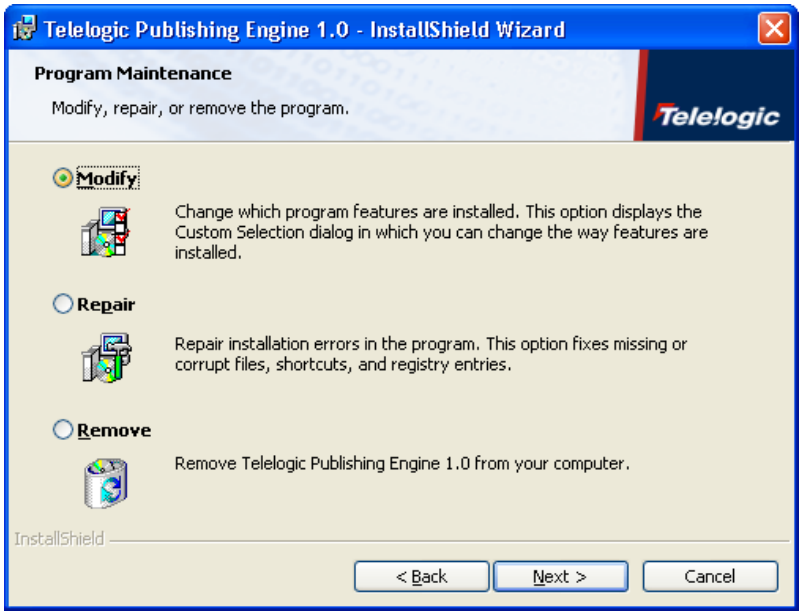


Figure 16

Once installed you need to activate the addin in Tau. Activating the addin can be done from Tau's addins page of the Customize function.

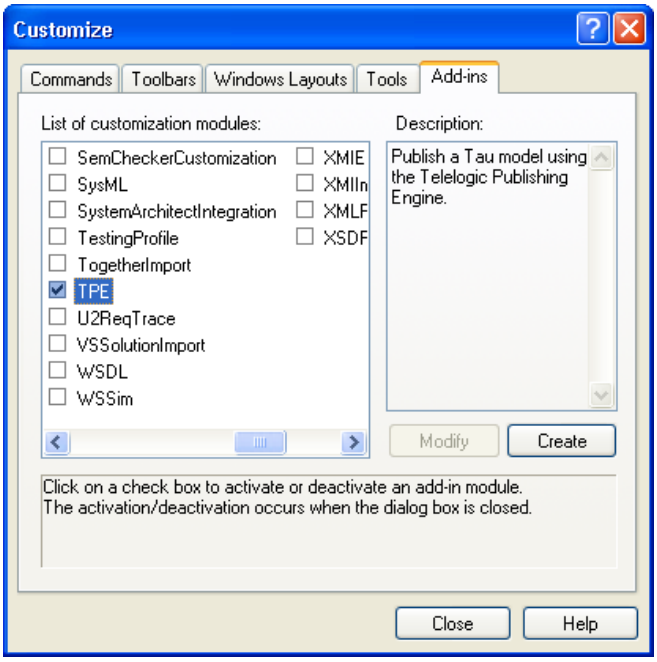


Figure 17

NOTE You need to have a Tau model loaded for the TPE addin to be listed.

Shared Document Library

The TPE Tau addin, much like the DOORS, makes use of the concept of a shared document template library. The content of this shared library (which is a folder on the file system, usually on a shared network drive) is displayed to the user in the Publish wizard.

NOTE As Tau does not have a server side component, the shared document library path is stored in the TPE_DOCUMENT_LIBRARY system variable. The variable is setup by the TPE installer to point to the Tau examples folder in the TPE installation. You can change this value to the desired location once TPE is installed.

Usage

Start the TPE addin from the Tools->Publish… menu.

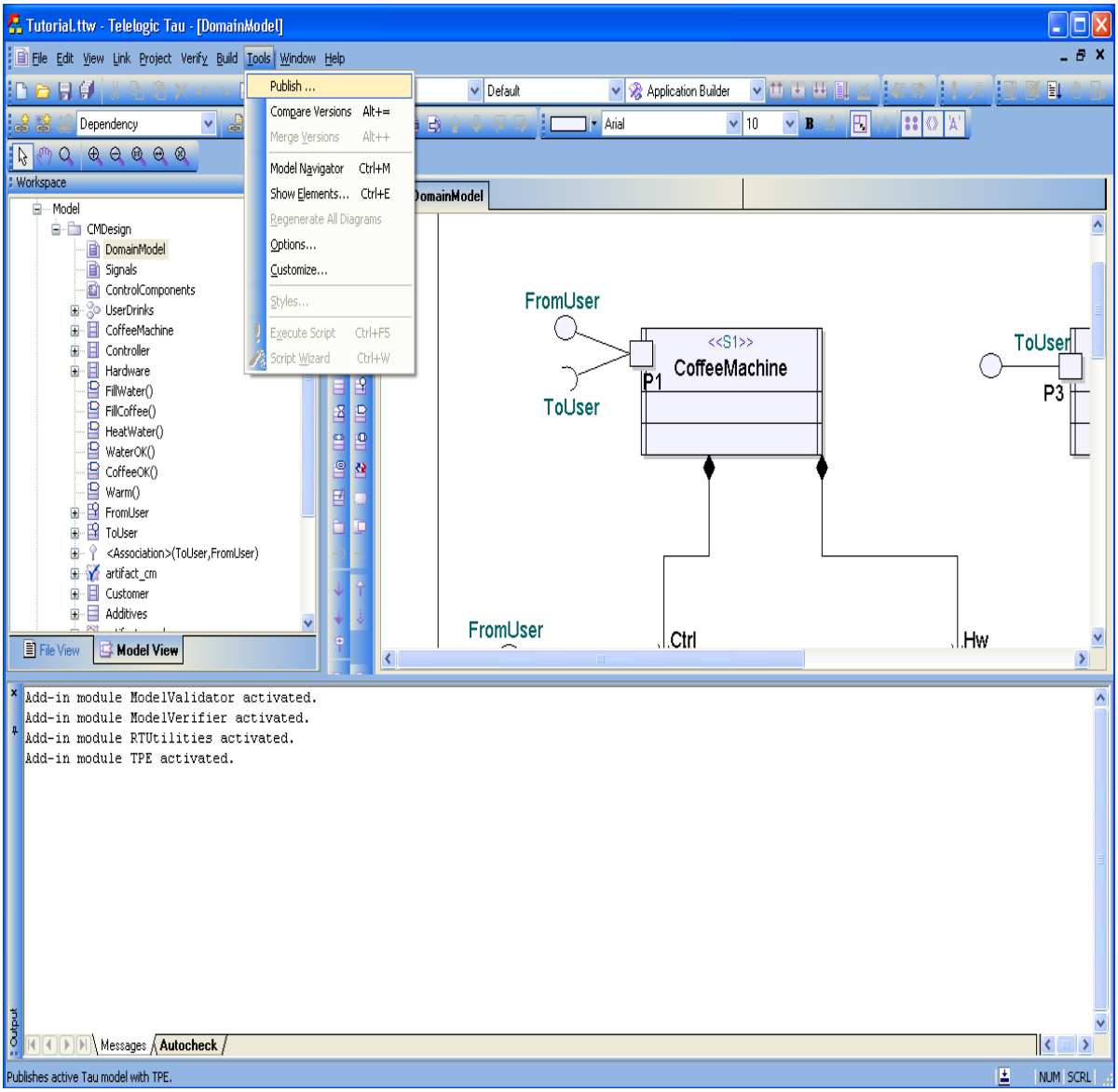


Figure 18

The TPE Publish Wizard is started.

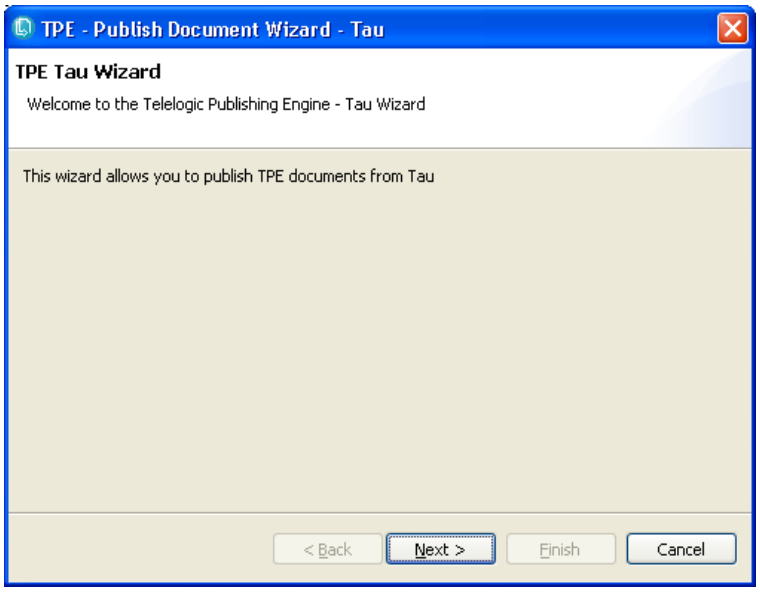


Figure 19

Select Next.

The content of the TPE Document Library is listed

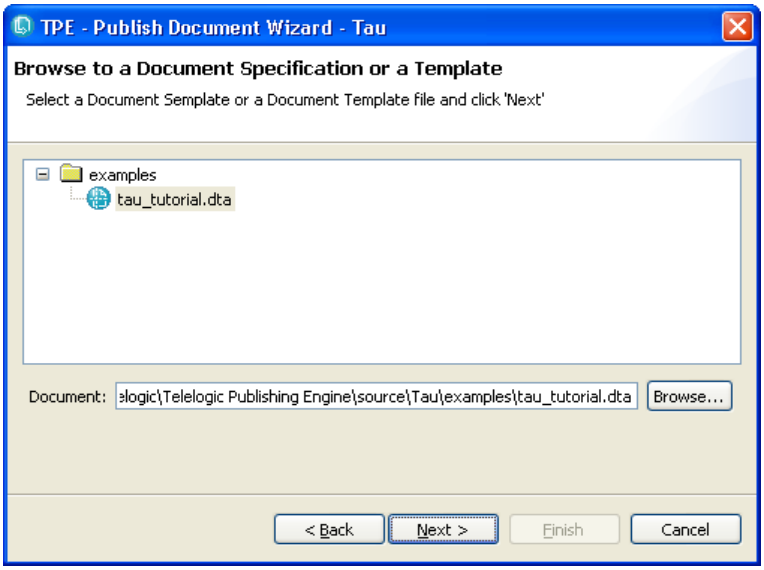


Figure 20

Select the desired document template/specification

NOTE The tree is populated with all the document templates and specifications found in the Shared Document Library.

The list of data sources for the current template is displayed

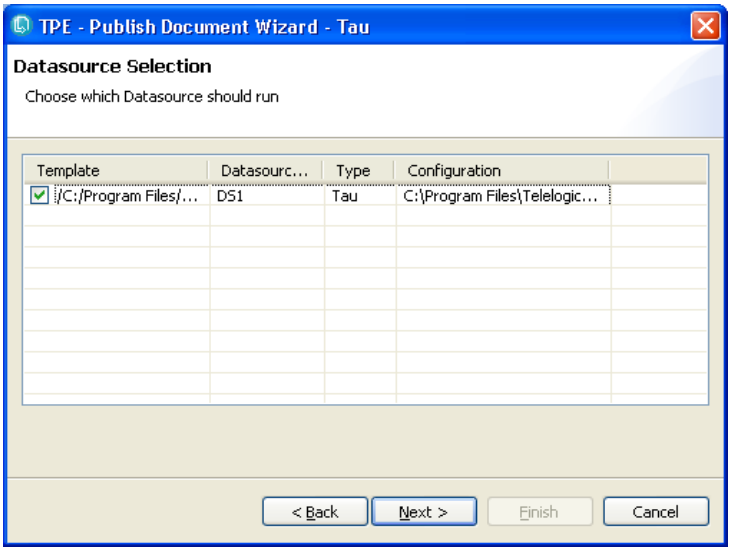


Figure 21

NOTE Unchecking a data source will mark it as ignored. All the queries belonging to ignored data sources are discarded when producing the output document.

The current Tau project is assigned by default to all the Tau data sources in the template. You can change the association as needed using the browse button available in the Configuration column.

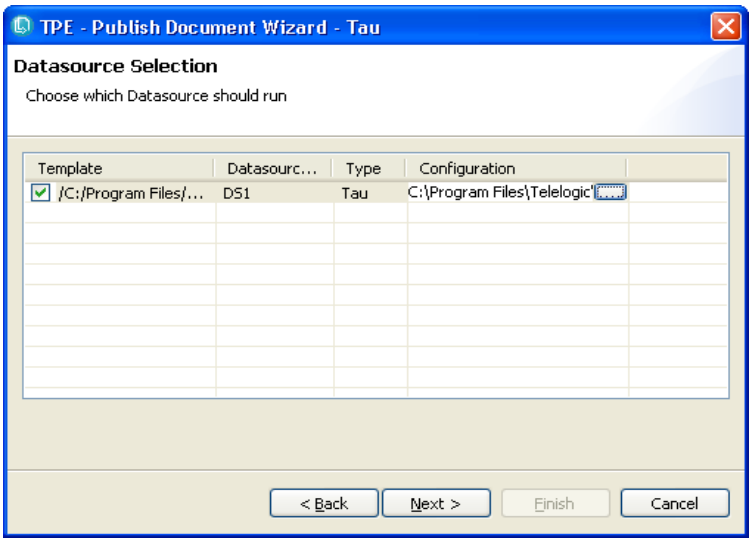


Figure 22

Select next.

You can now select the output types you are interested in.

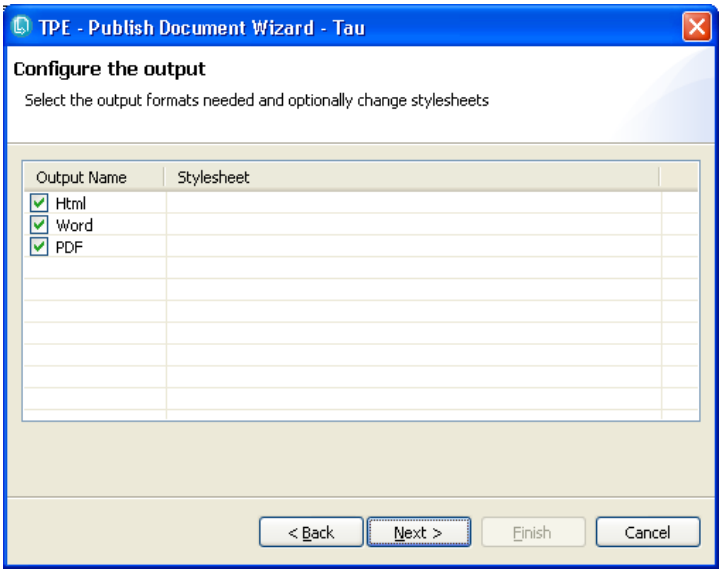


Figure 23

For any output type you can select the stylesheet to use.

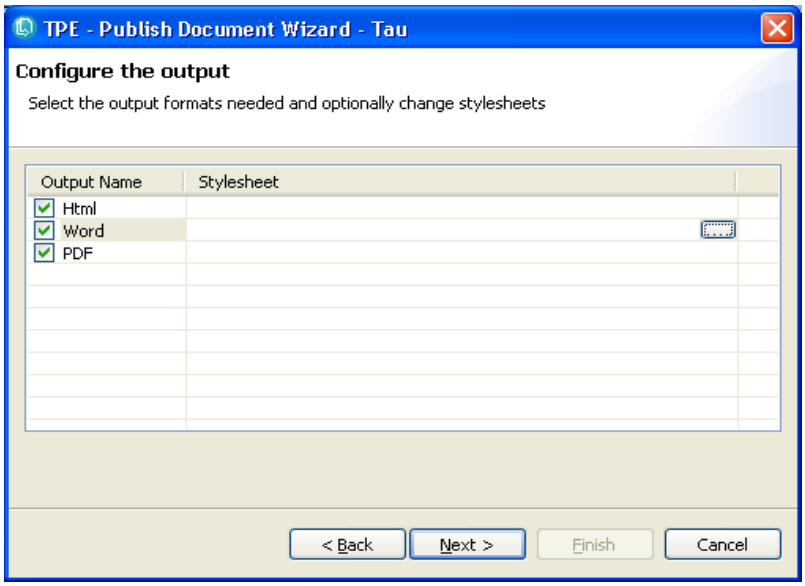


Figure 24

Select the Word stylesheet provided with TPE.

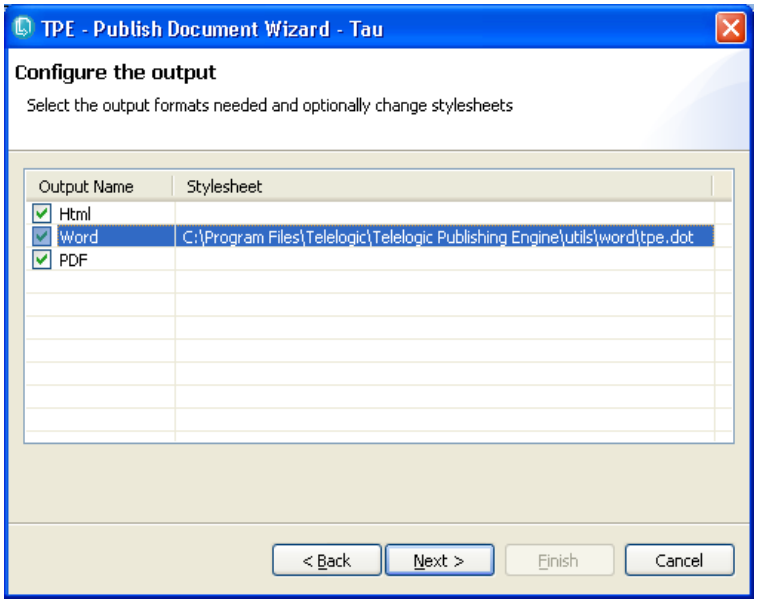


Figure 25

Choose the runtime options.

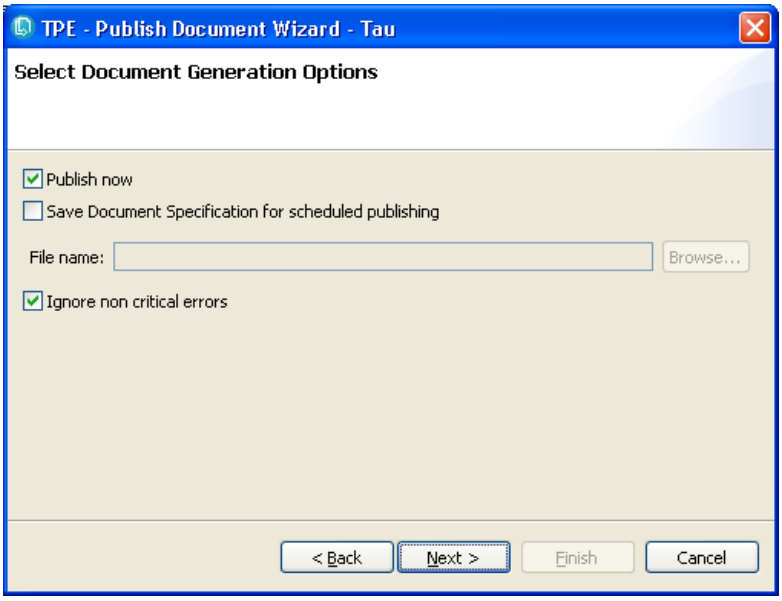


Figure 26

Property	Description
Publish now	If unchecked the document generation is not started
Save document specification for scheduled publishing	Checking this flag will allow you to select a location where the document specification will be saved. The document specification reflects all the options you've made in the wizard.
Ignore non-critical errors	If checked it will instruct TPE to proceed with the document generation if not every data source is properly configured.

Appendix: Notices

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.

Trademarks