SysML Validation Extension

Disclaimer

Version 2.0.x of the *SysML Validation Extension* has been successfully tested for deployment with EA version 15.x.

This deployment, as well as these guidelines, may or may not be applicable to any later version of the tool as released by the vendor, Sparx Systems. If required, updates to this software will be made available to support future versions of Enterprise Architect.

There is no guarantee that versions prior to EA 15.x will work properly. No effort will be made to support earlier releases of Enterprise Architect.

Great care has been taken during development to use SQL statements that are supported across the common backend database platforms. Nonetheless, should a statement fail to execute correctly, please refer to the Troubleshooting part of this User Guide for assistance.

If any other problems are encountered, either during installation or operation of this software, please contact us through any of the channels listed at the bottom of this document.

Dependencies

The add-in depends on the following components being installed on the system:

- Interop.EA.dll (part of the standard Sparx installation files).
- Microsoft .Net Framework 4 Client Profile.

Limitations of the trial version.

The following limitations apply to the trial version:

- The software activation is granted for eight (8) consecutive days.
- Only the first five (5) rule matches for any rule are reported back to the User,
- A maximum of 200 rule matches are reported for the duration of the trial period.

Supported Rule Sets

A global property for each rule configuration file can be set to exclude all Packages with a particular Status value (see the screenshot below). This allows filtering out sandbox or other non-production folders from the result set.

Note that all validations are performed at the Package level. Sub-Packages are automatically included unless filtered out.

The rules are split into separate files relating to specific SysML model sections being validated: Activity, Block Definition elements, Bock Definition features, Internal Block, Requirements, Interaction (Sequence), and Use Case. In addition a “General” rule set validates aspects of the model not specific to SysML.

A special” ALL_RULES” file runs all of the rule files in sequence.

Example of a rule specification in the Activity file:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE Validations >
<Validations Version="5" RunSilent="False" VerboseLevel="3" IgnorePackageWithStatus="">
    <ACT Description="Activity Diagram Validations">
      <Elements>
        <ObjectsWithNoClassifier Enabled="True" ID="PartitionWithNoClassifier" Description =
        "Verify Activity Partitions have a Classifier defined">
          <ElementType Name="'ActivityPartition'" MatchFilterCriteria="True" />
          <ElementMetatype Name="" MatchFilterCriteria="True" />
        </ObjectsWithNoClassifier>
```

Note the option for the entire file to:

- Ignore Packages with a specific status value.
- Run the validation in silent mode (i.e. no prompts).

While the structure of a rule varies from rule category to category, some of the consistent properties are:

Enabled	Set it to “False” to disable the rule.
ID	Unique identifier of the rule. When creating custom rules, assign them unique values.
Description	Displays in the SysML Validator output tab in the System Output window as the rules are executed.
Selection criteria	Varies by rule. The “MatchFilterCriteria” option indicates (when TRUE) if the search runs as specified (e.g. select elements where type = ‘ActivityPartition’) or (when FALSE) if the query is negated (e.g. select elements where type <> ‘ActivityPartition’).

Activity rules

Rule ID	*PartitionWithNoClassifier*
Description	Verify that Activity Partitions have a Classifier defined.
Notes	Includes checking if the specified Classifier exists in the current repository.

Rule ID	*CallBehaviorActionWithNoClassifier*
Description	Verify Call Behavior Actions have a Classifier defined.
Notes	Checks if the Call Behavior Action is actually invoking an Activity, Interaction or State Machine. Includes verifying if the specified Classifier exists in the current repository.

Rule ID	*CallOperationActionWithNoClassifier*
Description	Verify Call Operation Actions have a Classifier defined.
Notes	Checks if the Call Operation Action is actually invoking an Operation. Includes verifying if the specified Classifier exists in the current repository.

Rule ID	*SendSignalActionWithNoClassifier*
Description	Verify Send Signal Actions have a Signal Classifier defined.
Notes	Checks if the Send Signal Action has a reference to a Signal element. Includes verifying if the specified Classifier exists in the current repository.

Rule ID	*ActionPinWithNoClassifier*
Description	Verify Action Pins have a Classifier defined.
Notes	Includes checking if the specified Classifier exists in the current repository. By default, Action Pins named ‘target’, automatically created by EA for simulation purposes, are filtered out.

Rule ID	*ActivityParameterWithNoClassifier*
Description	Verify Activity Parameters have a Classifier defined.
Notes	Includes checking if the specified Classifier exists in the current repository.

Rule ID	*ActionsWithNoParentPartition*
Description	Verify every Action is a child of a Partition.
Notes	None

Rule ID	*AcceptEventWithNoTrigger*
Description	Verify Accept Event Actions have a Trigger defined.
Notes	Includes checking if the specified Trigger exists in the current repository.

Rule ID	*IncompatibleActivityParametersAndActionPins*
Description	Verify Call Behavior Actions Pins are compatible with the properties of the corresponding Activity Parameters.
Notes	Validates that: • Every Action Pin has a corresponding Activity Parameter. • The multiplicity values are the same. • The stereotype values are the same. • The data type/classifier values are the same. • The direction, exception and stream properties are the same.

Rule ID	*IncompatibleConnectedActionPinClassifiers*
Description	Validate connected Action Pins have compatible Classifiers.
Notes	Relates to Action Pins connected together via Object Flow. By default the code verifies if the two Classifiers (when not the same) are nonetheless compatible by means of a Generalization hierarchy.

Rule ID	*ActionPintoActionPinConnectors*
Description	Validate only Object Flow connects Action Pins together.
Notes	None

Rule ID	*ActionConnectors*
Description	Validate only Control Flow connects Actions (as input or output).
Notes	None

Rule ID	*ActionFlowToAllocatedBlocks*
Description	For each source element of a pair of Actions connected via Control Flow, and where the Actions are allocated to different Blocks via Activity Partition containment, verify that its allocated Block has a Part property, the type of which corresponds to the Block typing the Partition containing the target Action.
Notes	The intent is to check if Block-to-Block interactions, as specified in Activity diagram Control Flow connectors, are also reflected at the Block Definition level.

Rule ID	*ActionPinFlowToAllocatedBlocks*
Description	For each source element of a pair of Action Pins connected via Object Flow, and where the parent Actions of the Pins are allocated to different Blocks via Activity Partition containment, verify that its allocated Block has a Part property, the type of which corresponds to the Block typing the Partition containing the target Pin's parent Action.
Notes	The intent is to check if Block-to-Block interactions, as specified in Activity diagram Object Flow connectors, are also reflected at the Block Definition level.

Block Definition rules

Rule ID	*BlockAssociationEndProperties*
Description	Verify for each (navigable) connector end that role name and multiplicity are set.
Notes	Checks that for both Reference Association and Part Association connectors the role name and multiplicity properties defined.

Rule ID	*PartsWithNoClassifier*
Description	Ensure Parts (of a Block) have a Classifier defined.
Notes	Includes checking if the specified Classifier exists in the current repository.

Rule ID	*PortsWithNoClassifier*
Description	Ensure Ports (of a Block) have a Classifier defined.
Notes	Includes checking if the specified Classifier exists in the current repository.

Rule ID	*FullPortClassifier*
Description	Ensure Full Ports (of a Block) are classified by a Block (not an InterfaceBlock).
Notes	None.

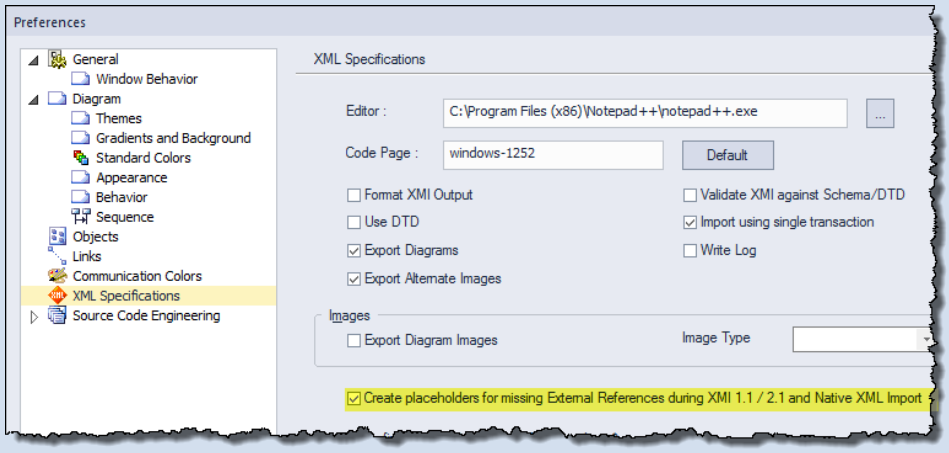
Feature related rules

Rule ID	*UnresolvedAttributeDataType*
Description	Report Attributes with unresolved Classifiers.
Notes	Verifies the Attributes defined in Signal elements. Includes checking if the specified Classifier exists in the current repository.

Rule ID	*UnresolvedBlockOperation*
Description	Report unclassified Operations (i.e. with an unresolved return type).
Notes	Verifies the Operations defined in Block and InterfaceBlock elements. Includes checking if the specified Classifier exists in the current repository.

Rule ID	*UnresolvedBlockOperationParameter*
Description	Report Operations with unresolved parameter Classifiers.
Notes	Verifies the Parameters of Operations defined in Block and InterfaceBlock elements. Includes checking if the specified Classifier exists in the current repository.

General (non SysML specific) rules

Rule ID	*UnresolvedConnectorEnds*
Description	Locates relationships in which either the source or the target element is missing from the repository. Missing connector end references typically occur during the import (manually or through version control) of incomplete models from other repositories. **Caveat:** the rule can only detect this condition if the *Create Placeholders for Missing External References* property is set! 
Notes	None

Rule ID	*OrphansWithNoConnectors*
Description	Similar to the Enterprise Architect "orphan report" (elements not in any diagram) but also verifies that the element is not referenced in other ways. WARNING: On a large dataset execution can take time!
Notes	By default, Action Pins named 'target' or 'result', automatically created by EA for simulation purposes, are filtered out. Verifies that the element: • Has no connectors. • Is not used as a Classifier. • Is not used as a Trigger. • Is not used as a reference Tagged Value.

Internal Block rules

Rule ID	*IncompatibleConnectedPortClassifiers*
Description	Validate connected Ports have compatible Classifiers.
Notes	By default the validation takes into account the compatibility of: • The direction of Flow Properties (if any). • The direction of Directed Features (if any). Comparison between Properties of two Classifiers is based on: 1) A matching Property name for Directed Properties. 2) A matching Property Classifier for Flow Properties and Directed Features. Comparison between Directed Operations of two Classifiers is based on the concatenation of Operation name and return type being the same.

Requirement rules

Rule ID	*UnsatisfiedRequirements*
Description	Verify Requirements are satisfied/realized using a connector type that is SysML compliant.
Notes	By default, the rule specifically checks for the SysML *Satisfy* relationship targeting the Requirement. The source of the relationship can be any element.

Rule ID	*UntestedRequirements*
Description	Verify Requirements are verified/tested using a connector type that is SysML compliant.
Notes	By default, the rule specifically checks for the SysML *Verify* relationship targeting the Requirement from the SysML Test Case element.

Sequence Interaction rules

Rule ID	*SequenceMessageNoOperation*
Description	Report Sequence messages with no associated Operation.
Notes	An option is provided to filter out Sequence messages: • Targeting one or more element types. • Targeting one or more element stereotypes.

Rule ID	*SequenceMessageMissingOperation*
Description	Report Sequence messages with an unresolved Operation reference (e.g. the Operation has been deleted from the model).
Notes	An option is provided to filter out Sequence messages: • Targeting one or more element types. • Targeting one or more element stereotypes.

Rule ID	*SequenceWithNoClassifier*
Description	Verify Sequence elements (Lifelines) have a Classifier defined.
Notes	None.

Use Case rules

Rule ID	*UseCaseToUseCaseConnectors*
Description	Verify Use Case to Use Case connector types are SysML compliant.
Notes	Checks if the connector type is either *UseCase* or *Generalization*.

Rule ID	*UseCaseToUseCaseStereotypes*
Description	Verify Use Case to Use Case connector stereotypes are SysML compliant.
Notes	Checks if the connector stereotype is either *include* or *extend*.

Rule ID	*ActorToUseCaseConnectors*
Description	Verify Actor to Use Case connector types are SysML compliant.
Notes	Checks if the connector type is either *UseCase* or *Association*.

Rule ID	*ActorWithNoDescription*
Description	Report Actor elements with no description/notes.
Notes	None.

Rule ID	*UseCaseWithNoDescription*
Description	Report Use Case elements with no description/notes.
Notes	None.

Rule ID	*DuplicateActorName*
Description	Report duplicate Actor elements.
Notes	Actors should be defined one time in the repository (typically in a library) and reused wherever needed. This facilitates traceability.

Rule ID	*MissingUseCaseToRequirement*
Description	Report Use Cases with no SysML compliant relationship to one or more Requirements.
Notes	Specifically check for the SysML *Refine* relationship from the Use Case to the Requirement. Type and Stereotype filters are available at the Use Case and Requirement level.

Installation

The installation process is the same for both the trial and the full version.

First, **exit any running instances of Enterprise Architect**, then launch the “setup.exe” program and follow the on-screen instructions.

The installation will attempt to update the Windows registry, so the User needs to ensure that s/he has sufficient privileges to run the setup program.

The recommended install path is to place the DLL and any supporting files in an *Addins* folder in the Sparx Systems installation directory, e.g.

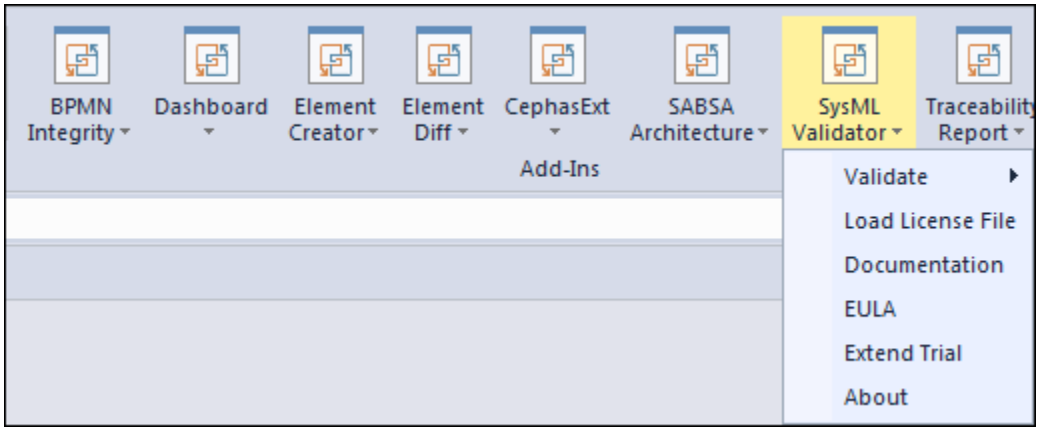
C:\Program Files (x86)\Sparx Systems\Addins.

Note that older versions of the software are automatically removed and replaced.

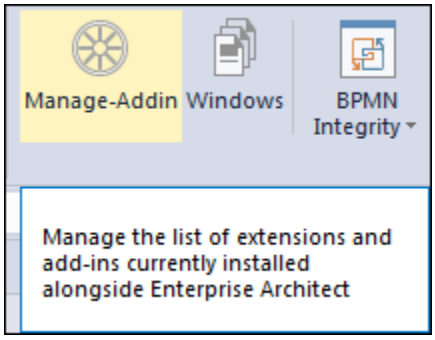
Should the installation fail for any reason other than insufficient User privileges, please take appropriate screenshots and email the data to the support address listed at the bottom of this document.

Verifying the installation

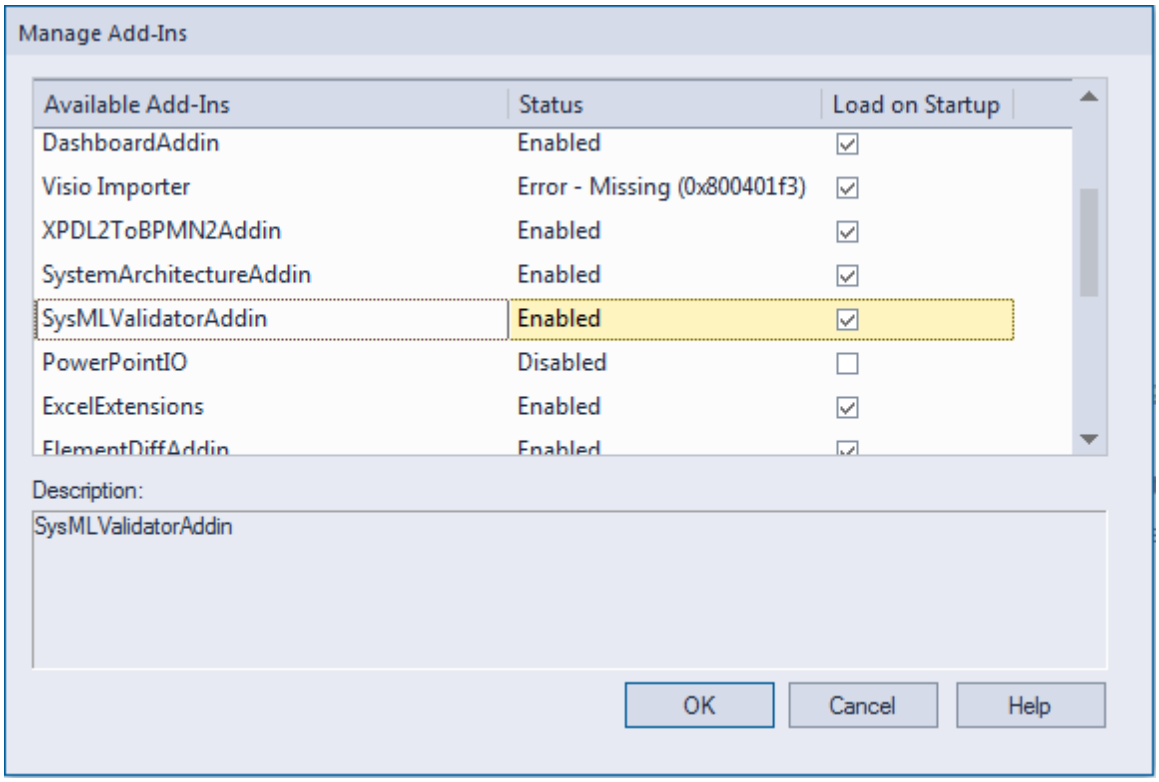
Bring up Enterprise Architect, without necessarily opening a repository, and verify that the *SysML Validator* extension has been loaded using the *SPECIALIZE → Add-Ins* ribbon panel:



Should the extension not be present, use the *SPECIALIZE → Add-Ins* ribbon panel and select:



And confirm that the *SysML Validator* extension is loaded and enabled:



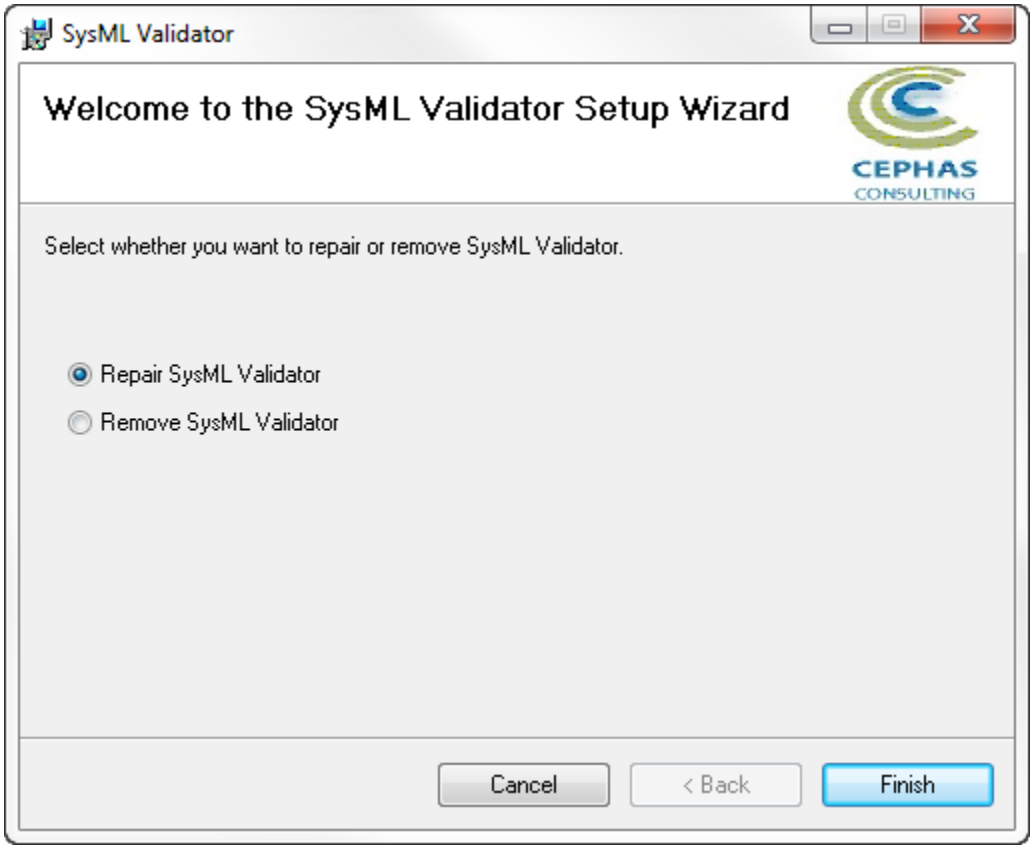
If an error status is shown, as in the example above for the *Visio Importer*, this typically means that either:

- The installation process failed and that the DLL cannot be located in the Windows registry, or in the file system.
- The installation did succeed but the DLL file was later moved or deleted.

If the *SysML Validator* entry itself is not found then the extension installation did not complete successfully.

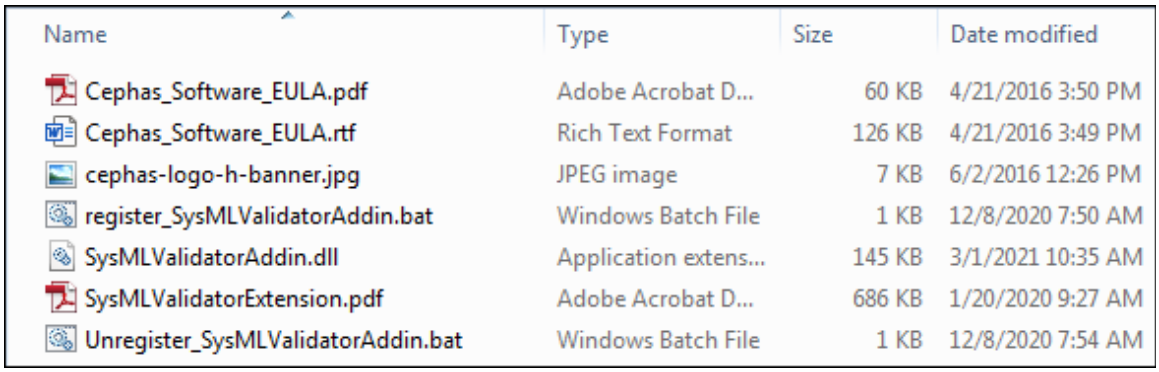
To fix an incorrect installation:

- Exit out of all instances of Enterprise Architect.
- Launch the setup process again. The installer will automatically provide a repair option:



If, after the repair procedure, the *SysML Validator* extension is still not loaded correctly in Enterprise Architect, remove the program through the Windows control panel and start the installation process over.

At the completion of a successful installation the following files are installed in the selected directory:



Name	Type	Size	Date modified
Cephas_Software_EULA.pdf	Adobe Acrobat D...	60 KB	4/21/2016 3:50 PM
Cephas_Software_EULA.rtf	Rich Text Format	126 KB	4/21/2016 3:49 PM
cephas-logo-h-banner.jpg	JPEG image	7 KB	6/2/2016 12:26 PM
register_SysMLValidatorAddin.bat	Windows Batch File	1 KB	12/8/2020 7:50 AM
SysMLValidatorAddin.dll	Application extens...	145 KB	3/1/2021 10:35 AM
SysMLValidatorExtension.pdf	Adobe Acrobat D...	686 KB	1/20/2020 9:27 AM
Unregister_SysMLValidatorAddin.bat	Windows Batch File	1 KB	12/8/2020 7:54 AM

Installing the license key file

Trial version

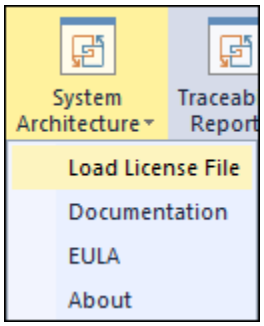
The software installation automatically loads the trial version license key.

Licensed version

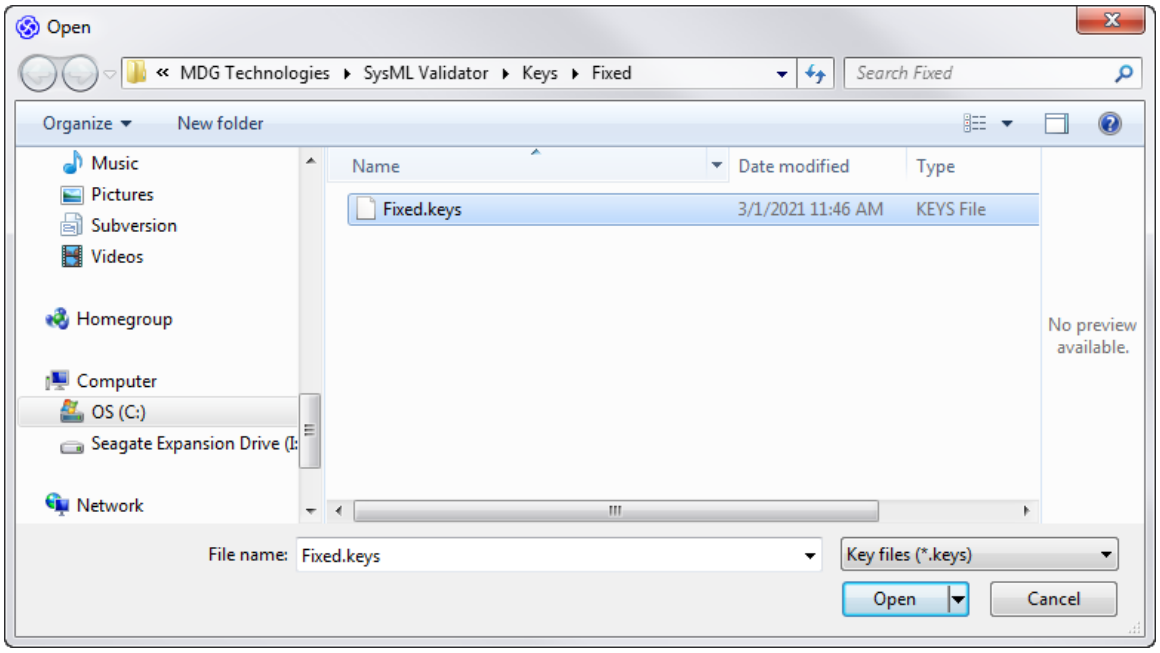
Once the full version of the product has been purchased, a *<license type>.keys* file will be provided by Cephas Consulting which needs to be installed **by each licensed User of the software**, even if a floating/shared or site license key is acquired.

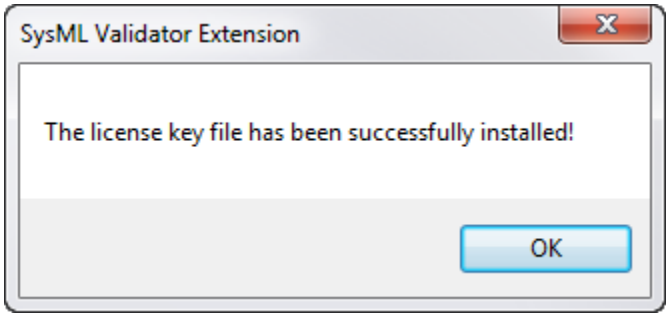
Caveat: the keys file can be saved anywhere on your system, but its **name should not be changed**.

To install the license key file, using the *SPECIALIZE → Add-Ins* ribbon panel, select:



Next, select the provided file from the folder in which you copied it. For example, for a fixed license key:





After installing the license key file, continue with the next section.

Adding the User license key

Trial version

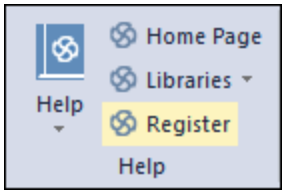
Skip this step if you are using the trial version of Enterprise Architect. Otherwise you can optionally register the trial key with EA.

Fixed and Site Licenses

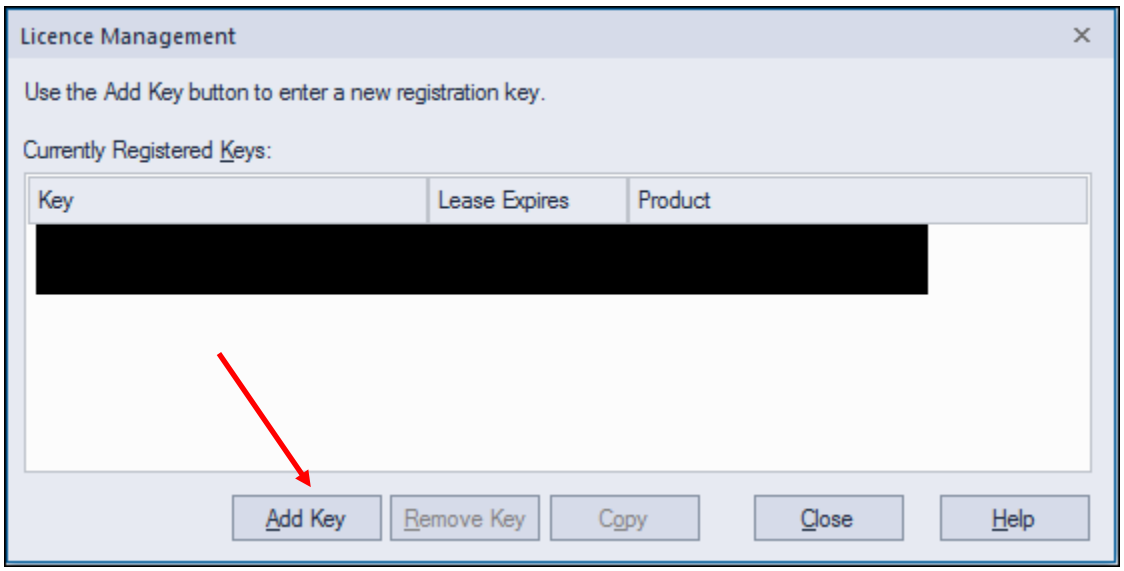
The following step is required for the fixed as well as the site license of the product in order to make Enterprise Architect verify the software license.

See here for how to install a floating/shared license.

Open the START ribbon and select:



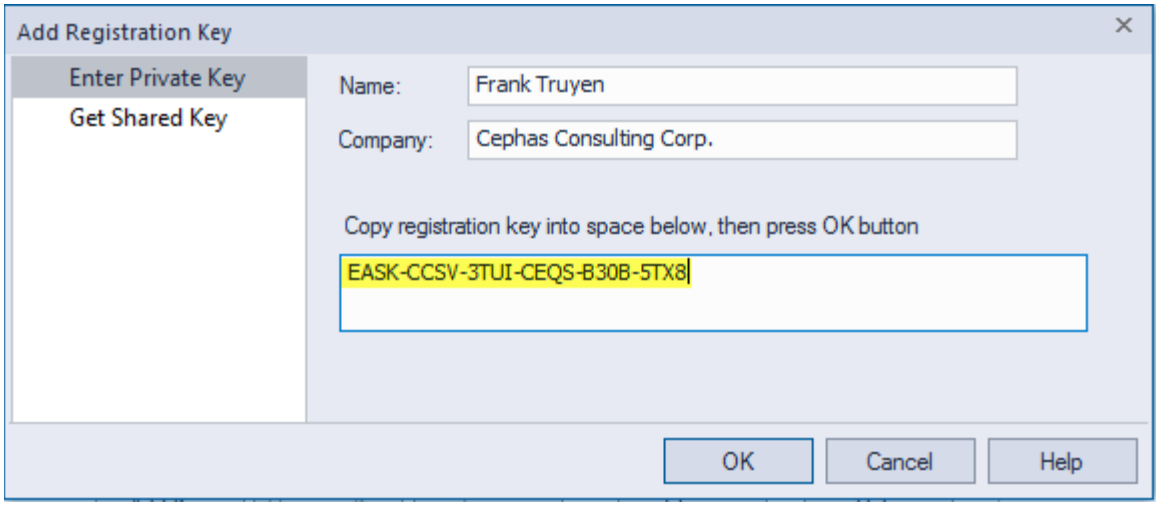
Next, click “Add Key”:



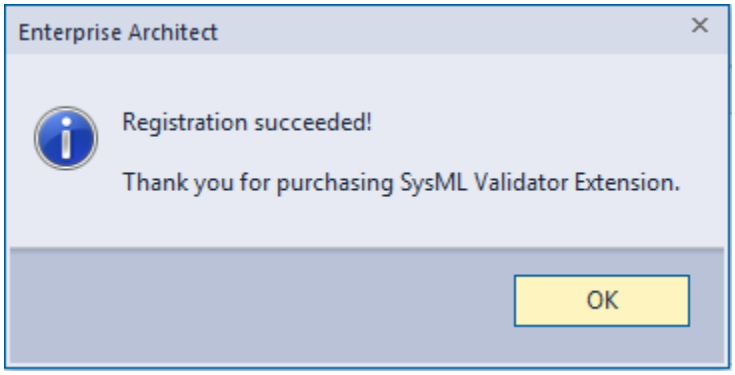
Enter or copy/paste, either the trial key:

EASK-CCSV-3TUI-CEQS-B30B-5TX8

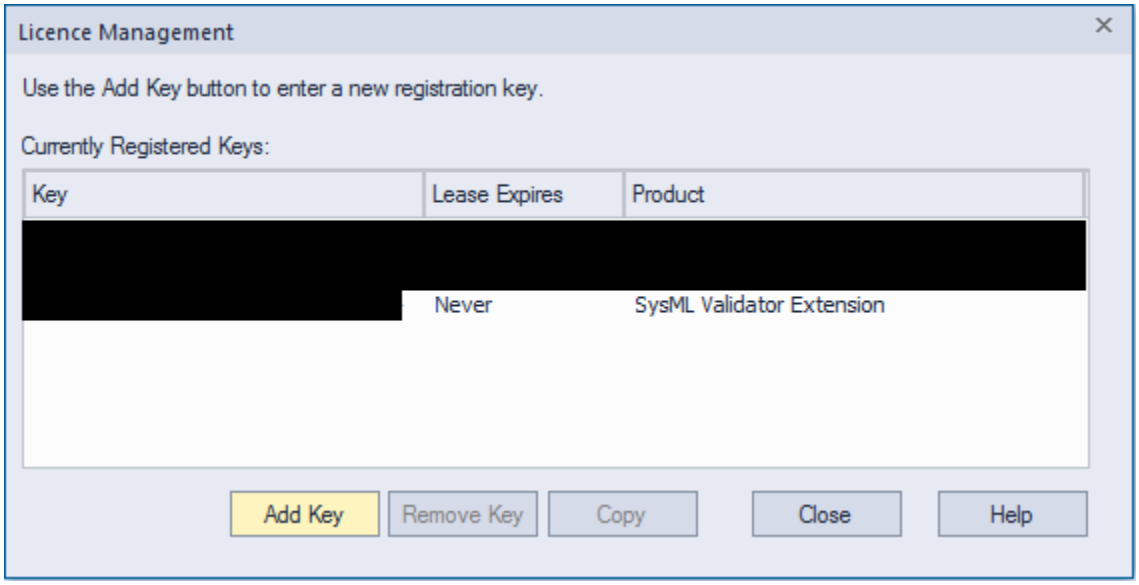
Or one of the fixed license keys provided as part of the software purchase.



Enterprise Architect will confirm the successful addition of a key:

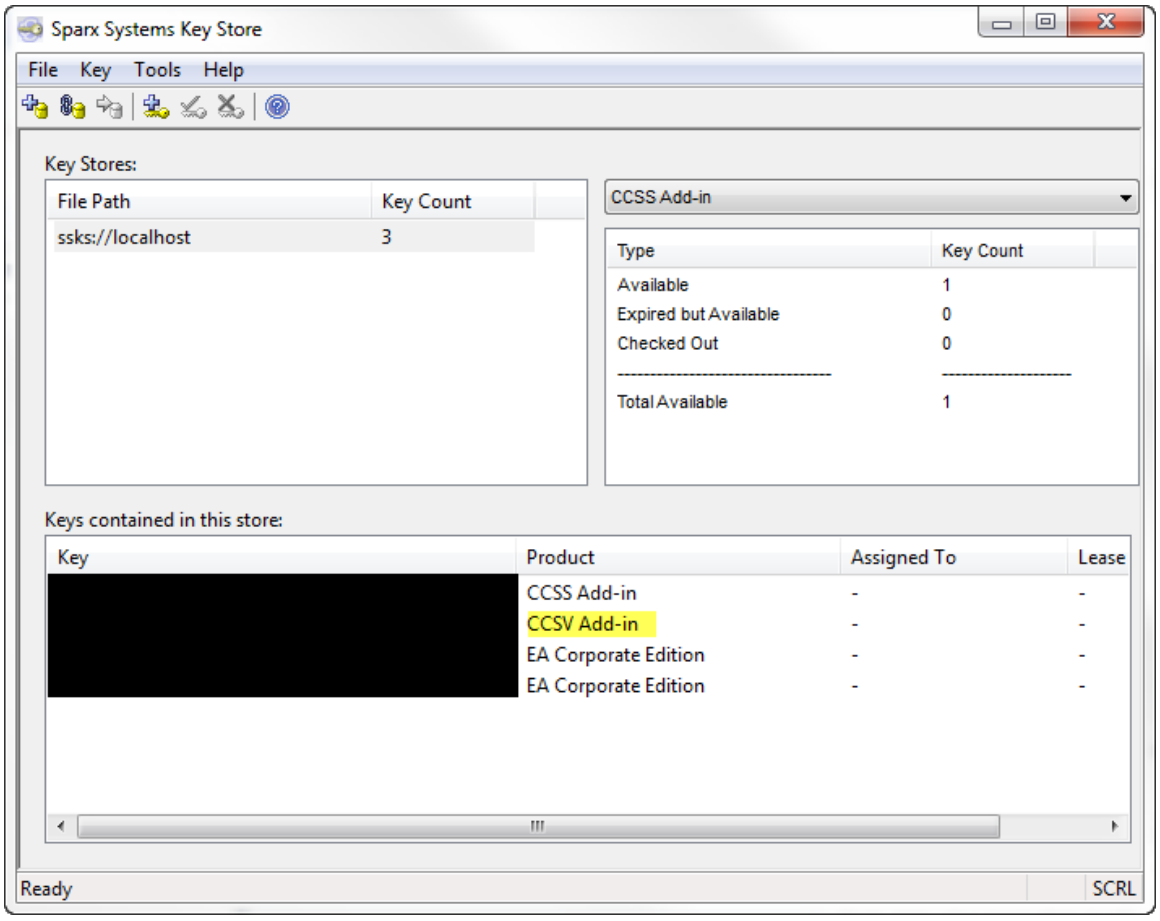


The license is now added to the registered keys:

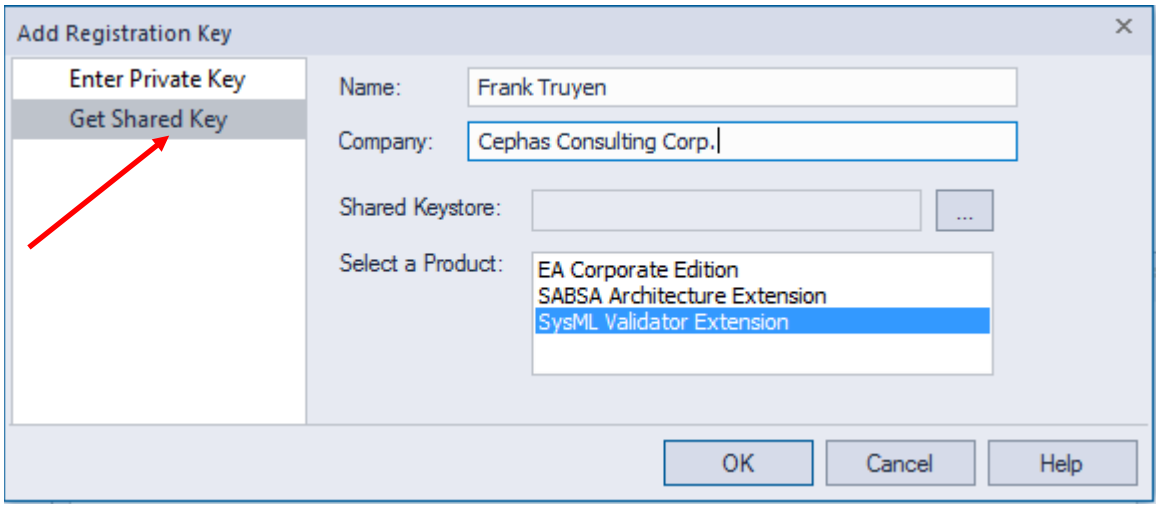


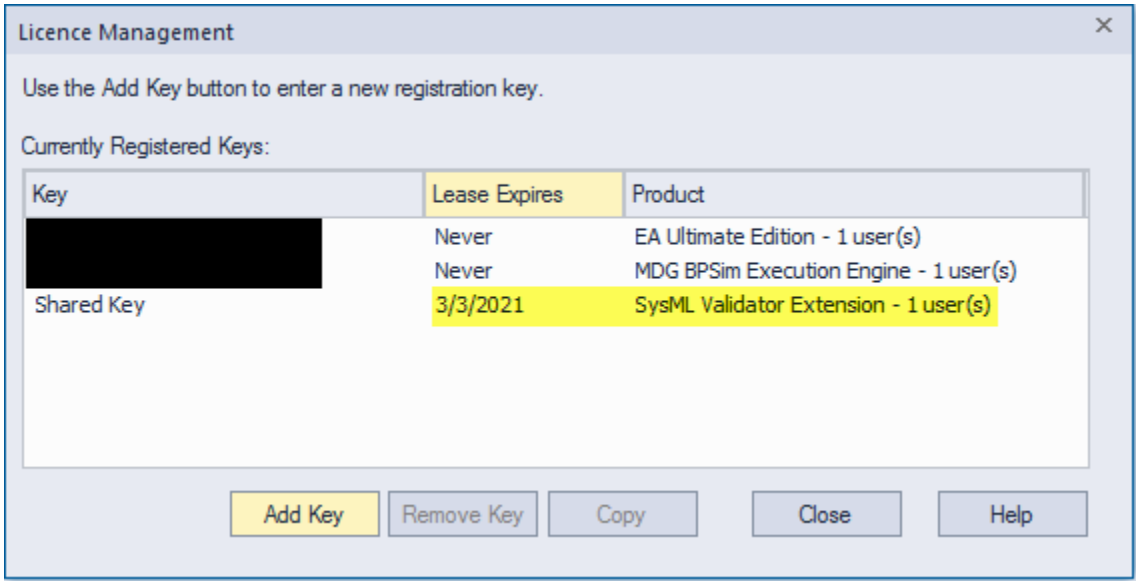
Floating/Shared Licenses

First the administrator needs to add the key/s to the Sparx System key store (**version 2.3 or higher**), using the same process as for Enterprise Architect license keys:



Individual Users can then obtain a key from the store using the “Get Shared Key” tab:

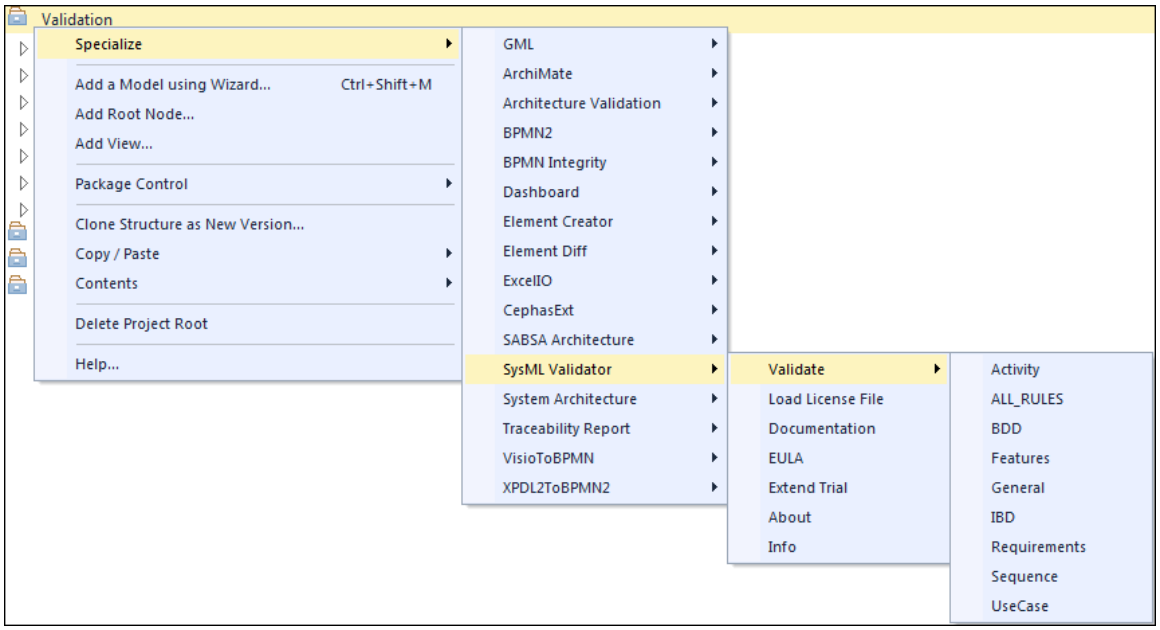



Licence Management
Use the Add Key button to enter a new registration key.
Currently Registered Keys:
Key
Lease Expires
Product
[redacted]
Never
EA Ultimate Edition - 1 user(s)
Never
MDG BPSim Execution Engine - 1 user(s)
Shared Key
3/3/2021
SysML Validator Extension - 1 user(s)
Add Key
Remove Key
Copy
Close
Help

Running the validation

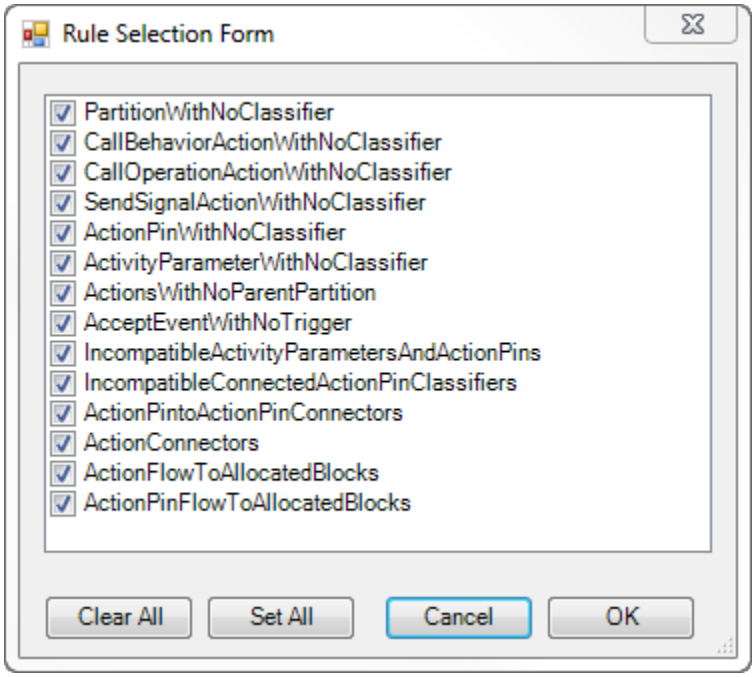
Any validation is performed in the context of a selected Package structure. All child Packages in the hierarchy (if any) are included by default (Packages can be filtered out by setting the *IgnorePackageWithStatus* property in the rule set).

Right click a Package, or the top of a Package hierarchy, and select any of the available Rule Sets:



Caveat: the larger the package hierarchy selected the more time the validation will take!

Unless the *RunSilent* option is enabled, a form displays when launching the validation for a specific rule set (except for “ALL_RULES”). For example:



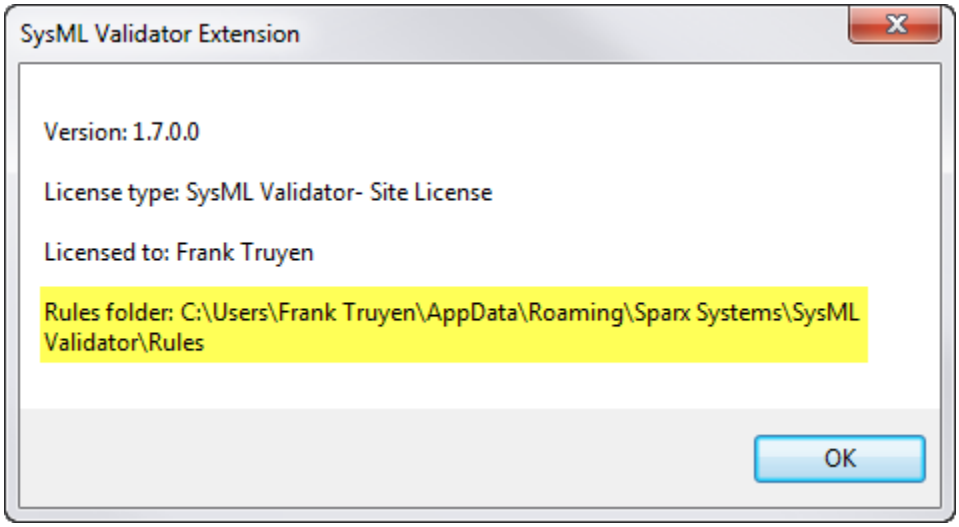
This allows rules to be enabled/disabled for a specific validation. To disable rules by default, create a custom rule set.

Customizing the Rule Set

Caveat: it is best not to modify the default Rule Sets!

Instead:

- Make a copy of the xml file **in the same** Rules folder. The location of that folder can be easily determined by selecting the "About" option of the extension menu:



- Optionally move the original xml file to an archive folder.
- Open the new file in any XML editor and make the required changes in that copy.

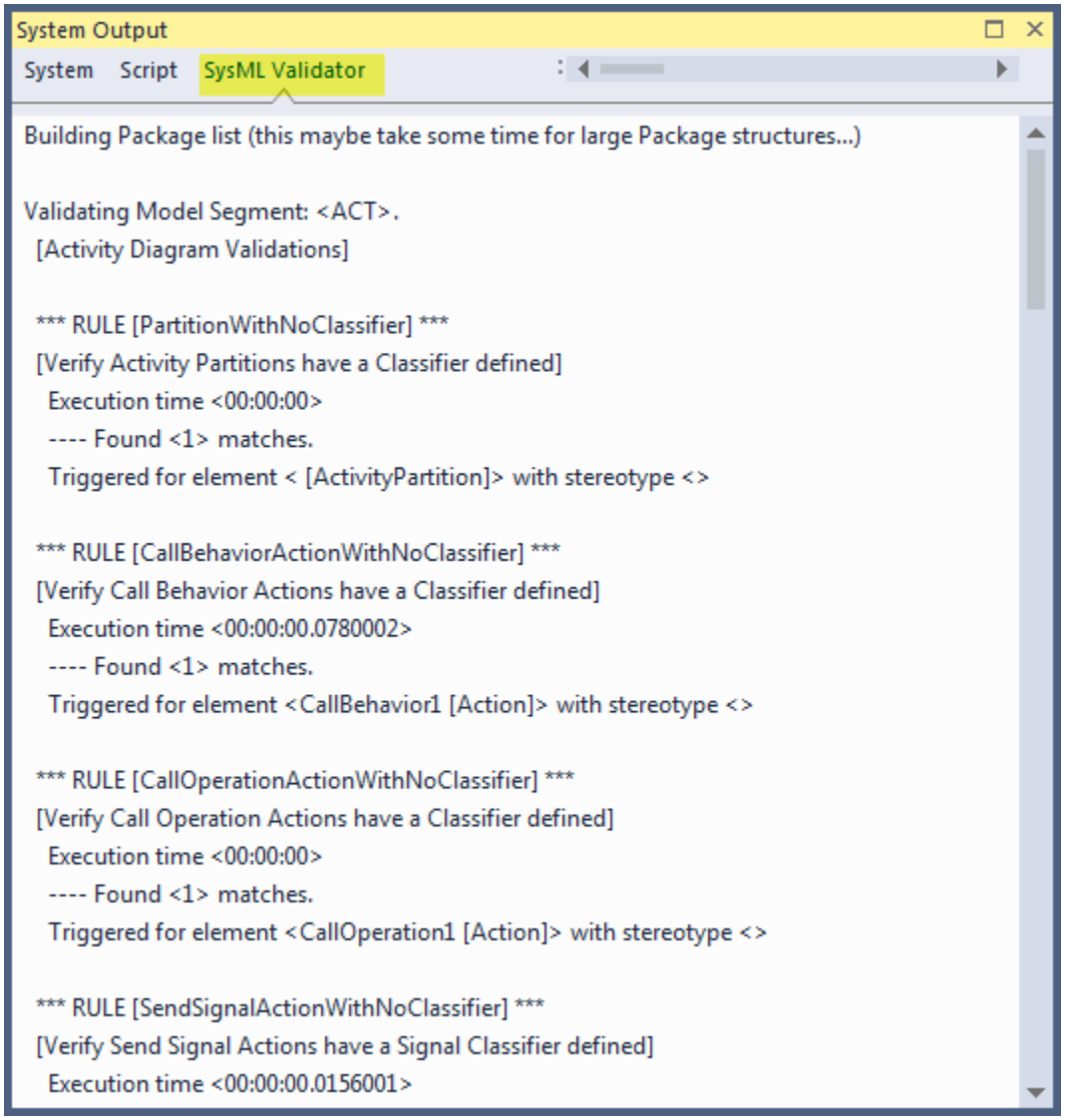
When adding a new Rule to a Set, ensure that it is given a unique ID (i.e. name) value.

Custom rule files are automatically detected and made available for selection.

Rule changes can be made in between validations, while EA is running!

Verifying the result set

During execution, the “SysML Validator” tab in the System Output window automatically opens and displays the results of the validation. For example:



Every rule match is listed in this window using the format:

> Triggered for <connector, element, Attribute, Operation or diagram name> {additional information}

For rules relating to connectors, the associated objects (source and/or target) and diagram/s are reported on additional output lines.

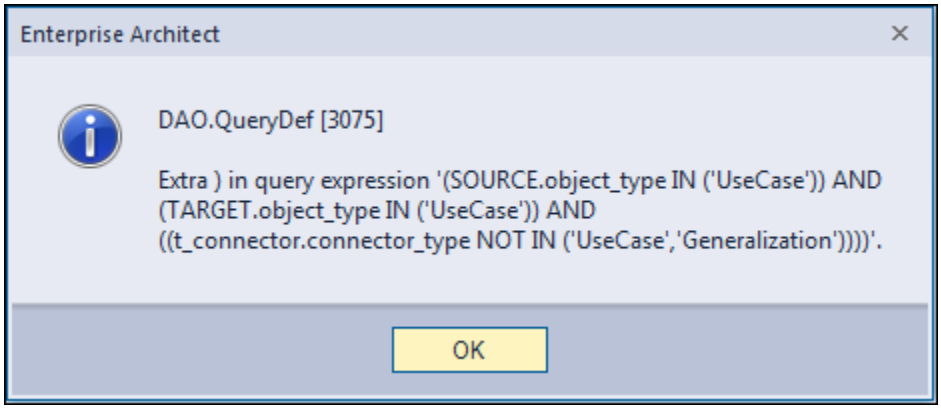
Where applicable, additional output lines may be added to include related diagrams.

Single click a line to automatically locate its related element, Attribute, Operation or diagram in the Browser (note that some element types in EA are not included in the Browser). For connector rules, click a related diagram (if provided) to automatically open it and select the connector in that diagram (you may need to scroll the window to see the selection).

Double click a line to open the element, Attribute or Operation properties, or to open the diagram associated with the rule.

Troubleshooting

Should a SQL statement fail to execute properly, Enterprise Architect will display an error message dialog similar to this:



As of version 15.x of Enterprise Architect this type of error is not relayed back to the application which is unaware that a problem occurred.

Please follow this procedure to report any error encountered:

- Take a screenshot of the error message.
- If the error is related to a failed database query, locate the DBError.txt file in %APPDATA%\Sparx Systems\EA and include it in your message.
- Before dismissing the error notification, look at the System Output window to determine the rule being executed at the time of the failure. For example:

  *** VALIDATING RULE [UseCaseToUseCaseConnectors] on connector type <'UseCase','Generalization'> ***

- If you are validating the repository using a customized rule set, please include that xml file in the data provided back to Cephas.
- Also provide:
  - Your database type (Microsoft Access, SQL Server, Oracle, etc.) and version number.
  - The version of the SysML Validator extension by selecting the “About” option of the extension menu.
  - The version of Enterprise Architect being used.
  - Your operating system and any other execution environment information that may be relevant.

Support and contact information

Use the contact information below for any installation or runtime issues with the extension.

Feature requests or suggestions for improvement are also welcome!

Contact: Frank Truyen

Email: support@enterprisemodelingsolutions.com

Phone : 208-462-4863.