

MBSE Style Guide Extension

MBSE Style Guide Extension	1
Disclaimer	2
Dependencies	2
Limitations of the trial version	2
Supported rule sets	3
Example of rule specifications in the Activity XML file:	3
Example (State Machine) rule set	4
Activity Style Guide rules	5
Block Definition Style Guide rules	5
State Machine Style Guide rules	8
Change Trigger and Guard Condition Definition Helper	13
Installation	18
Verifying the installation	20
Licensing	23
Trial version	23
Licensed version	23
Request the license activation	23
Receive the license activation	24
Running the Style Guide validation	26
Package level validation	26
Diagram level validation	26
Element level validation	27
User Interface	27
Customizing the rule set	28
Verifying the result set	29
Troubleshooting	31
Support and contact information	32

Disclaimer

Version 2.5.x of the *MBSE Style Guide Extension* has been successfully tested for deployment with both the both 32- and 64-bit versions of Enterprise Architect 16.1. There is no guarantee that versions prior to EA 16.1 will work properly. No effort will be made to support earlier releases of Enterprise Architect.

The add-in, 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.

Great care has been taken during development to use SQL statements that are supported across the common database platforms, including:

- SQLite (QEA and QEAX Sparx Systems file based repositories).
- Firebird
- MySQL 8.0
- SQL Server 2019
- Oracle Express Edition 21c.

Note that Microsoft Access databases are NOT supported.

Should a database SQL 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 4.8.

Limitations of the trial version.

The following limitations apply to the trial version:

- The software activation is granted for five (5) consecutive days.
- Only the first five (5) matches for any rule are reported back to the User.
- Only the first twenty (20) rule violations are reported back to the User per Enterprise Architect session.
- The options to automatically fix up the model are disabled.

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 Packages from the result set.

Note that this filter option only applies to validations performed at the Package level (sub-Packages are automatically included unless explicitly filtered out).

The rules are split into separate XML files relating to specific SysML model sections being validated: [Activity](#), [Block Definition](#), and [State Machine](#).

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

Example of rule specifications in the Activity XML file:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE Validations >
<Validations Version="1" RunSilent="False" VerboseLevel="3" IgnorePackageWithStatus="" >
  <ACT Description="Activity Diagram Validations">
    <Elements>
      <ObjectsWithNoClassifier Enabled="True" ID="PartitionWithNoClassifier" Description = "Verify Activi
        <ElementType Name=" 'ActivityPartition' " MatchFilterCriteria="True" />
        <ElementMetatype Name="" MatchFilterCriteria="True" />
      </ObjectsWithNoClassifier>
      <ElementsWithInvalidParent Enabled="True" ID="ActionWithNoValidParent" Description = "Verify Activi
        <Element Type=" 'Action' " MatchFilterCriteria="True" />
        <Parent Type=" 'ActivityPartition', 'InterruptibleActivityRegion' " MatchFilterCriteria="False" />
      </ElementsWithInvalidParent>
    </Elements>
  </ACT>
</Validations>
```

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).
- Set the verbosity level of the ouput log.

While the structure of each rule may vary considerably, 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 MBSE Style Guide Validation output tab in the System Output window as the rules are executed.
Selection criteria	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').

Example (State Machine) rule set

```

1  <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2  <!DOCTYPE Validations >
3  <Validations Version="1" RunSilent="False" VerboseLevel="3" IgnorePackageWithStatus="">
4    <STM Description="State Machine rules">
5      <Elements>
6        <DuplicateName Enabled="True" ID="DuplicateStateName" Description = "Report
7          duplicate state element names inside a State Machine Region.">
8          <ElementType Name="'State'" MatchFilterCriteria="True"/>
9          <ElementStereotype Name="" MatchFilterCriteria="True" />
10         </DuplicateName>
11        <BlankStates Enabled="True" ID="BlankStates" Description = "Check for States with
12          no in/out transition behavior or triggers, guard conditions, or entry/do/exit">
13          <ElementType Name="'State'" MatchFilterCriteria="True"/>
14          <ElementStereotype Name="" MatchFilterCriteria="True" />
15         </BlankStates>
16        <AssignedBehaviors Enabled="True" ID="AssignedBehaviors" Description = "Check for
17          States with entry/do/exit behaviors that do not reference actual model behaviors">
18          <ElementType Name="'State'" MatchFilterCriteria="True"/>
19          <ElementStereotype Name="" MatchFilterCriteria="True" />
20         </AssignedBehaviors>
21        <InitialAndFinalStates Enabled="True" ID="InitialAndFinalStates" Description =
22          "Verify that for every State Machine Region one Initial and at least one Final
23          State are present">
24          <ElementType Name="'StateMachine', 'State'" MatchFilterCriteria="True"/>
25          <ElementStereotype Name="" MatchFilterCriteria="True" />
26         </InitialAndFinalStates>
27      </Elements>
28      <Connectors>
29        <OutgoingStateTransition Enabled="True" ID="OutgoingStateTransition" Description =
30          "Check for States with duplicate out transitions with the same guard and/or trigger"
31        >
32          <Source Type="'State'" Stereotype="" />
33          <Target Type="'State'" Stereotype="" />
34        </OutgoingStateTransition>
35        <GuardAndTrigger Enabled="True" ID="GuardAndTrigger" Autofix="False" AutoDisplay=
36          "False" Description = "Validate that guard conditions and triggers are specified in
37          a compliant way">
38          <Source Type="'State'" Stereotype="" />
39          <Target Type="'State'" Stereotype="" />
40        </GuardAndTrigger>
41      </Connectors>
42    </STM>
43  </Validations>

```

Activity Style Guide rules

Rule ID	<i>PartitionWithNoClassifier</i>
Description	Verify that Activity Partitions have a SysML Classifier defined.
Notes	Includes checking if the specified Classifier exists in the current repository.
Rationale	Ensure the Actions and Decisions included in a Partition are properly allocated to a structural element that is capable of performing that behavior.

Rule ID	<i>ActionWithNoValidParent</i>
Description	Verify every Action of an Activity is a child of either a Partition or an Interruptible Region.
Notes	Interruptible Regions can span multiple Partitions. EA uses a parent-child relationship between a Region and its content, similar to the parent-child relationship between a Partition and its content. This makes it more difficult to trace the Actions in a Region back to the underlying Partition (if any).
Rationale	Ensure the Actions of an Activity are allocated to a parent element that can itself be allocated to a structural element.

Block Definition Style Guide rules

Rule ID	<i>BlockAssociationEndProperties</i>
Description	Verify for each (navigable) connector end that role name and multiplicity are set.
Notes	Checks that for both SysML Reference Association and Part Association connectors the role name and multiplicity properties are defined.
Rationale	Enables: <ul style="list-style-type: none"> the ability to refer to downstream Properties of a Block using a navigable expression language (e.g., a dot notation like <i>role_name.role_name.property_name</i>). removes the ambiguity of determining whether the relationship is optional (0..1) or required (1..).

Rule ID	<i>BlockGeneralizations</i>
Description	Verify that a specialized Block adds value to the generalized Block.
Notes	Checks if the specialized Block has Parts and/or Operations/Receptions. Parts can be the result of adding Properties to the Block as well from adding Reference Association and/or Part Association connectors.
Rationale	Specialized Blocks that do not add semantic meaning should be removed from the model in order to simplify it.

Rule ID	<i>CleanBlockName</i>
Description	Ensure Block names are in camel case and have no spaces.
Notes	An auto-fix option is available to automatically update the Block names. It is disabled by default.
Rationale	Ensure model consistency by using a naming convention.

Rule ID	<i>CleanPropertyName</i>
Description	Ensure Property names are in lower camel case with _ separators and have no spaces.
Notes	An auto-fix option is available to automatically update the Property names. It is disabled by default. If the Property/Part is a result of a Reference Association or Part Association connector, then the auto-fix option also updates the corresponding connector role name.
Rationale	Ensure model consistency by using a naming convention.

Rule ID	<i>ValueProperty</i>
Description	Ensure Value Properties have a description and default value.
Notes	Verifies that Block Properties typed by a SysML Value Type have both a description (notes in EA) and a default value defined.
Rationale	<p>Value Properties generically represent a quantifiable structural feature of the Block. Its default value is often overridden at the instance level (or through redefinition).</p> <p>Ideally the unit-of-measure of the Value Type should (directly or indirectly via inheritance) resolve to a primitive (Integer, Real, Boolean, String, etc.) to facilitate simulation.</p> <p>A description is needed to help the developer understand the usage context of the Property.</p>

Rule ID	<i>CleanOperationName</i>
Description	Ensure the names of Operations and Receptions (including their Parameters) are in lower camel case with _ separators, and have and no spaces in them.
Notes	<p>Verifies the Operations and Receptions defined in Block and Interface Block elements.</p> <p>An auto-fix option is available to automatically update the names. It is disabled by default.</p> <p>For Receptions, the auto-fix option also updates the names of the corresponding Signals and their Attributes.</p>
Rationale	Ensure model consistency by using a naming convention.

Rule ID	<i>OperationImplementationSpecified</i>
Description	Ensure an Operation has either behavior or (pseudo-)code specified.
Notes	Verifies that the non-abstract Operations defined in a SysML Block have either code (Code tab in the Properties window) or pseudo-code (free text in Behavior tab), or a reference to another behavioral element - Interaction, Activity, or State Machine - (reference in Behavior tab).
Rationale	An Operation must be backed by something more concrete than its signature and documentation, either as input to the developer or a simulation engine, or as a source of code generation.

State Machine Style Guide rules

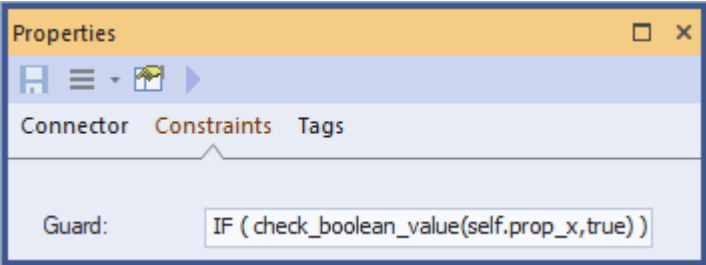
Rule ID	<i>DuplicateStateName</i>
Description	Verify there are no duplicate State element names inside a State Machine Region.
Notes	None.
Rationale	Each State name should be unique (within a Region) to ensure there is no ambiguity when referring to a current State using its path within the State Machine graph.

Rule ID	<i>BlankStates</i>
Description	Verify that there are no States without: <ul style="list-style-type: none"> • Incoming or outgoing Transition behavior or Triggers, or Guard conditions • no entry/do/exit actions.
Notes	None.
Rationale	Such States add no value to the model since they are simply passed through during State Machine simulation or execution.

Rule ID	<i>AssignedBehaviors</i>
Description	Check for States with entry/do/exit behaviors that do not reference actual model behaviors.
Notes	Verifies that entry/do/exit actions have either code (Code tab in the Properties window) or pseudo-code (free text in Behavior tab), or a reference to another behavioral element - Interaction, Activity, Operation, or State Machine - (reference in Behavior tab).
Rationale	A State action must be backed by something more concrete than its name and documentation, either as input to the developer or a simulation engine, or as a source of code generation.

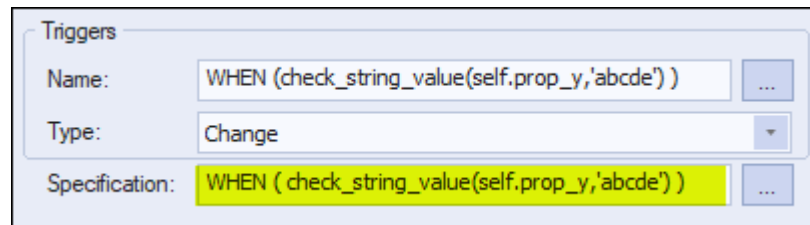
Rule ID	<i>InitialAndFinalStates</i>
Description	Verify that for every State Machine Region one Initial and at least one Final State are present.
Notes	None.
Rationale	This removes any ambiguity of how Regions are entered or exited by default, and also simplifies modeling concurrent Regions.

Rule ID	<i>OutgoingStateTransition</i>
Description	Check for States with duplicate out Transitions with the same Guard condition and/or Trigger/s.
Notes	None.
Rationale	No two Transitions out of a State should exhibit identical Trigger and Guard conditions. This would be an ill-formed State Machine, as each Transition would try to fire.

Rule ID	<i>GuardAndTrigger</i>
Description	Validate that Guard conditions and/or Triggers of a State Transition are specified in a Style Guide compliant way (see Notes for details).
Notes	<p>As an example, consider the following Block and its Operation definitions:</p> <div data-bbox="448 508 1382 940" style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <pre>bdd [package] Library [Library] «block» GuardAndChangeTriggerFunctions check_boolean_value(boolean_property: boolean, value: Boolean): boolean check_numeric_value_gt(numeric_property: Real, value: real): boolean check_string_value(string_property: String, string_value: string): boolean check_numeric_value_eq(numeric_property: Real, value: real): boolean check_numeric_value_lt(numeric_property: Real, value: real): boolean check_enum_value(enum_property: Enumeration, string_value: string): boolean</pre> </div> <p>This Block can be inherited by any other Block that owns a State Machine.</p> <p>Rules for a Guard condition:</p> <ul style="list-style-type: none"> • The text must be enclosed in a “IF (...)” expression. • The expression inside the parenthesis must map to a non-abstract Operation of the Block that resolves to a boolean result. • Any argument of the call that refers to a Property of the Block (owned or inherited) must be prefixed with the string “SELF.”. • The data types of the passed in arguments must be compatible with the data types of the Operation parameters. <p>For example (compare with the Operation definition above):</p> <div data-bbox="493 1446 1195 1709" style="border: 1px solid gray; padding: 10px; margin: 10px 0;">  </div> <p>Rules for Triggers of type Change:</p> <ul style="list-style-type: none"> • The text must be enclosed in a “WHEN (...)” expression. Refer to section 13.3.4 of the UML 2.5.1 specification for details.

- The expression inside the parenthesis must map to a non-abstract Operation of the Block that resolves to a boolean result.
- Any argument of the call that refers to a Property of the Block (owned or inherited) must be prefixed with the string “SELF.”.
- The data types of the passed in arguments must be compatible with the data types of the Operation parameters.

For example:

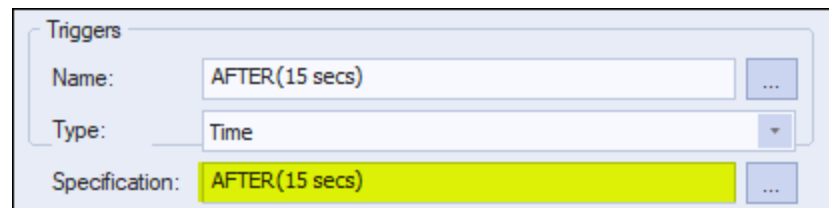


Rule for Triggers of type Time:

- The text must be enclosed in either:
 - a “After (...)” expression for a relative time event.
 - a “At (...)” expression for an absolute time event.

Refer to section 13.3.4 of the UML 2.5.1 specification for details.

For example:

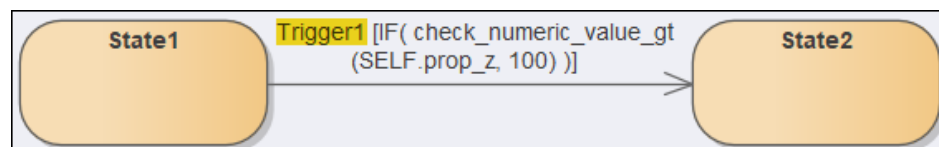


An auto-fix option is available to automatically enclose the current expressions. It is disabled by default.

A second auto-display option (disabled by default) ensures that the name of each Trigger element (by default, only the name of a Trigger is displayed by EA in a diagram label) matches the specification value. This option operates against all four types of Triggers.

For example, in the case of a Call Trigger:

Prior to applying the option:



Connector **Constraints** Tags

Guard:

Effect:

Effect is a Behavior

Triggers

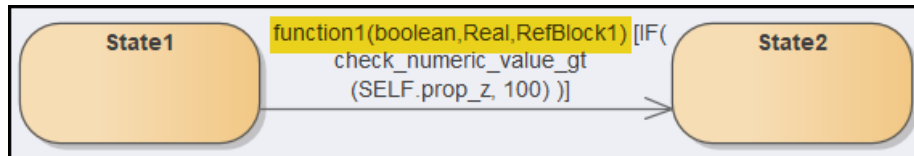
Name:

Type:

Specification:

Name	Type	Specification
Trigger 1	Call	ParentBlock.function1

After applying the option:



Triggers

Name:

Type:

Specification:

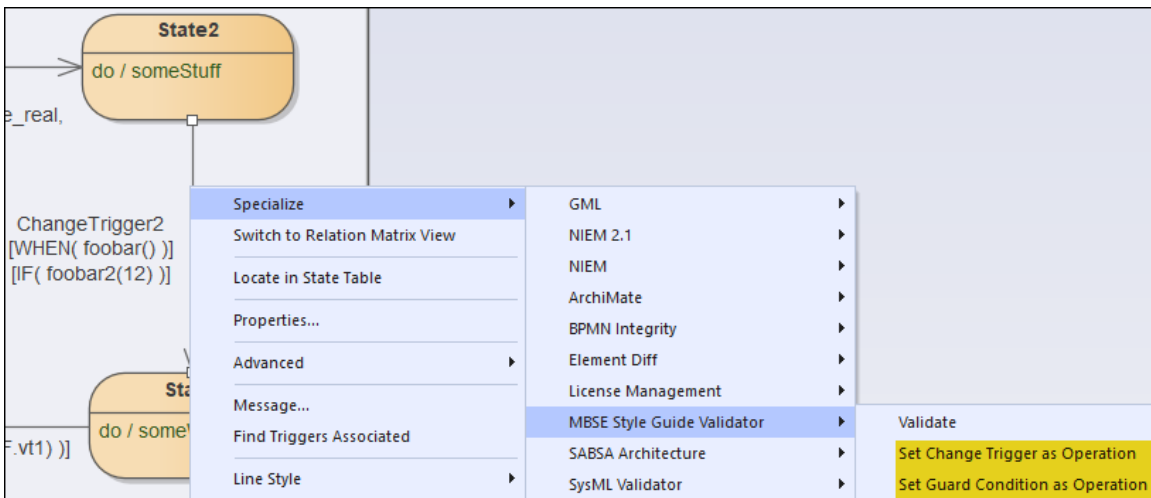
Note that if the combined value of all the trigger names exceeds 255 characters, the label text will be truncated.

Rationale	<p>To facilitate simulation and execution of State Machines, Change Triggers and Guard conditions should be represented by mathematically evaluable expressions that resolve to a boolean value.</p> <p>The easiest way to ensure this is to map them to non-abstract Operations of the Block owning the State Machine (either directly or via inheritance).</p>
------------------	--

	<p>For Triggers of type time, it is recommended that the modelers adopt a particular convention (e.g., all relative times shall be expressed in seconds).</p> <p>The enclosing naming conventions help ensure consistency across State Machines.</p>
--	--

Change Trigger and Guard Condition Definition Helper

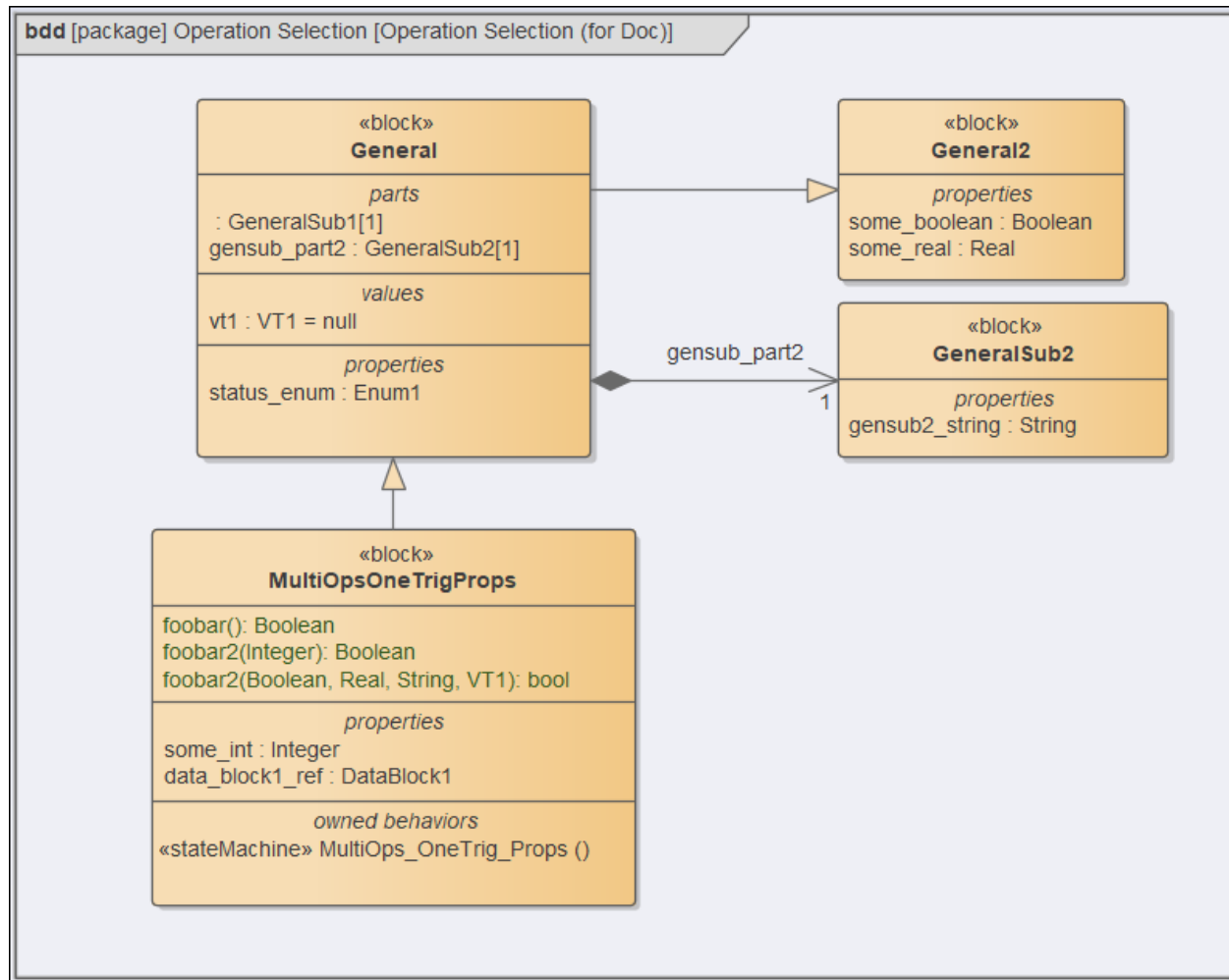
To facilitate mapping Change Trigger and/or Guard condition specifications to existing Operations of a Block (see rule *GuardAndTrigger* in the above table), right click the Transition connector and select either one of these two context menu options:



MBSE Style Guide

Either option sets the context for opening the dialog box shown on the next page.

The Operations and Properties available for selection in this example are derived from this model:



MBSE Style Guide

Trigger/Guard Condition Operation

Change Event Triggers: **1** Current Specification: **2**

Block Operations: **3**

Selected Operation: **4**

Current Argument: **5** **6** **7**

Argument Value Filtered by Argument Type:

Boolean: True False Numeric:

Text: **8** Use Empty Text

Property: **9**

Result Value: **10**

Status: **11**

Switch to Change Trigger/s Exit Apply

MBSE Style Guide

Field descriptions and User actions:

Field	Description
1.	Dropdown of defined Changed Triggers (if any) for this State Transition.
2.	The current specification (if any) for the selected Trigger or Guard condition. Read-only.
3.	Dropdown of all Operations defined by the Block owning the State Machine (including any inherited Operations), and which have a boolean return value.
4.	The currently selected Operation. Read-only.
5.	The currently selected argument (if any) of the Operation. Use the Next and Previous buttons to assign a value to each argument.
6.	When enabled, click Next to assign a value to the next argument (if any) of the current Operation. A valid value must be assigned to the current argument before switching to the next one.
7.	When enabled, click Previous to assign a value to the previous argument (if any) of the current Operation. A valid value must be assigned to the current argument before switching to the previous one.
8.	Depending on the data type of the current Operation argument, assign either a boolean, numeric or string static value. To pass in an empty string value, checkmark "Use Empty Text". To instead assign the dynamic value of a Block property, ignore these fields and select a Property (if available) from the Property dropdown (see 9).
9.	Depending on the data type of the current Operation argument, Properties may be available for selection from the dropdown (see example of a string property in the image above). The list of Properties is derived from the Block owning the State Machine (including any inherited

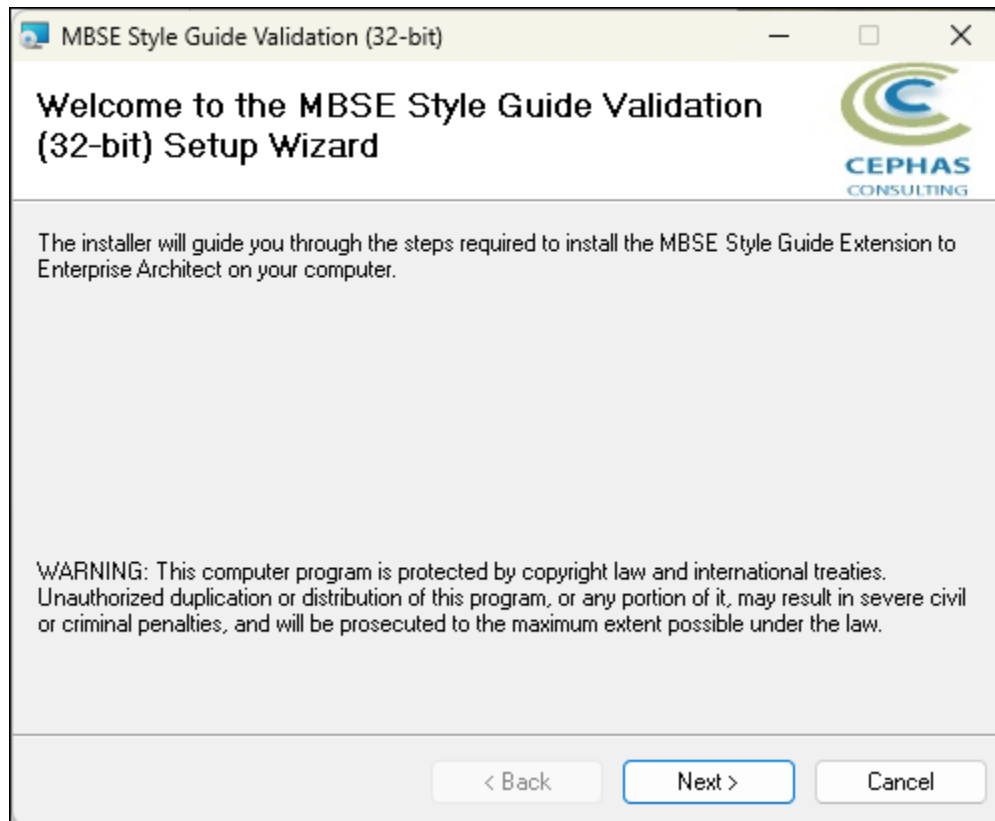
MBSE Style Guide

	<p>Properties), plus the Properties of Blocks it is associated with via Reference and Part Associations (traversed recursively across the break down structure). The name of each Property in the dropdown reflects its access path. Refer to the model shown here.</p> <p>For primitives (boolean, string, number), the value assigned to the Operation argument can be either a static value (see 8) or the dynamic value of a Property. To switch back from an assigned Property to a static value, select “None” from the dropdown. For non-primitive values, a Property must be selected (i.e. the “None” option will not be available).</p>
10.	Shows the specification value as it is being built. That value will be assigned to either the Change Trigger or Guard condition when the Apply button (see 14) is clicked. Read-only.
11.	Displays status messages such as help and error notifications. Read-only.
12.	Allows switching context between editing the specification of a Guard condition and that of one or more Change Triggers (if any are defined for the current Transition connector). Partial updates made in either context are saved before the switch.
13.	Click Exit to leave the interface. Any unsaved (un-applied) changes to either context are lost!
14.	<p>Click Apply to update either the currently selected Change Trigger or Guard condition specification. This button is only enabled:</p> <ul style="list-style-type: none"> - After values have been assigned to all arguments of an Operation. - If the new specification (see 10) is different from the current specification.

Installation

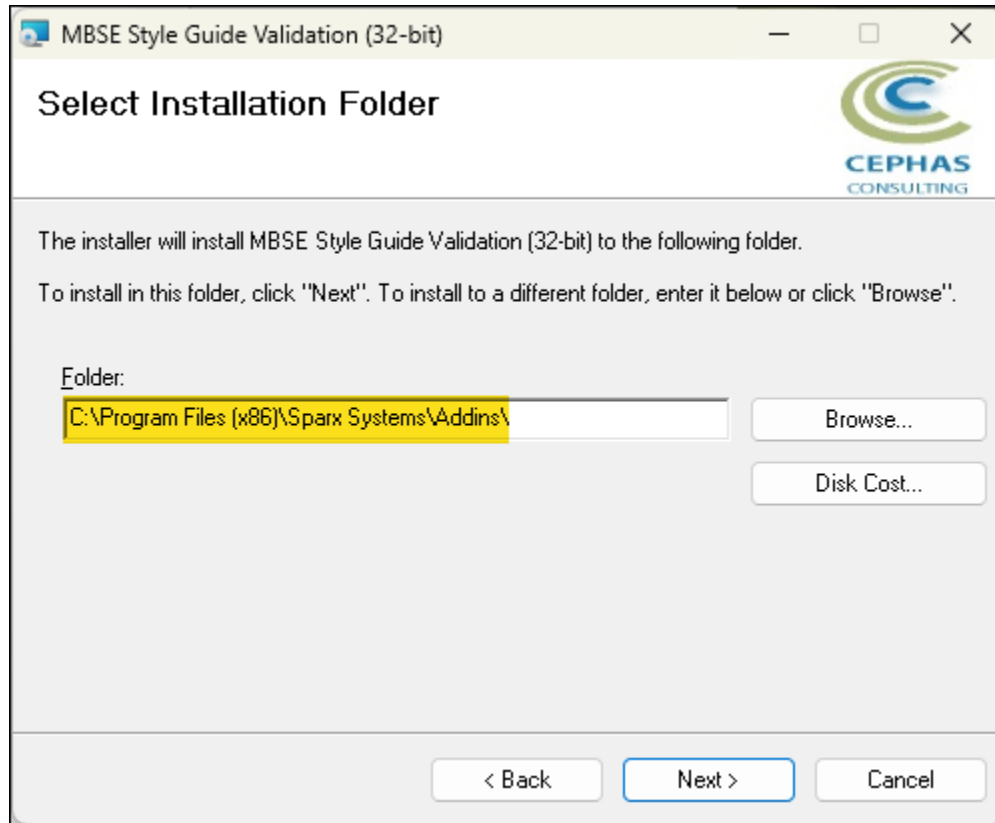
The installation process is the same for both the trial and the full version.

First, **exit any running instances of Enterprise Architect**, then launch the “setup.exe” program and follow the on-screen instructions. For example, for the 32-bit version:



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., for the 32-bit version:

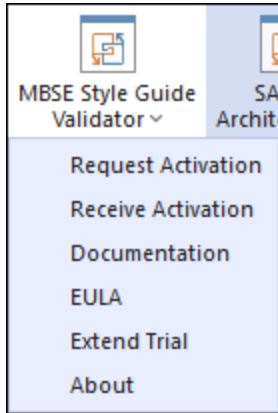


Note that older versions of the software are automatically removed and replaced.

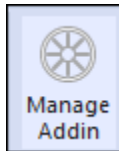
Should the installation fail for any reason other than insufficient User privileges, please take appropriate screenshots and email the data to the [support](#) address listed at the bottom of this document.

Verifying the installation

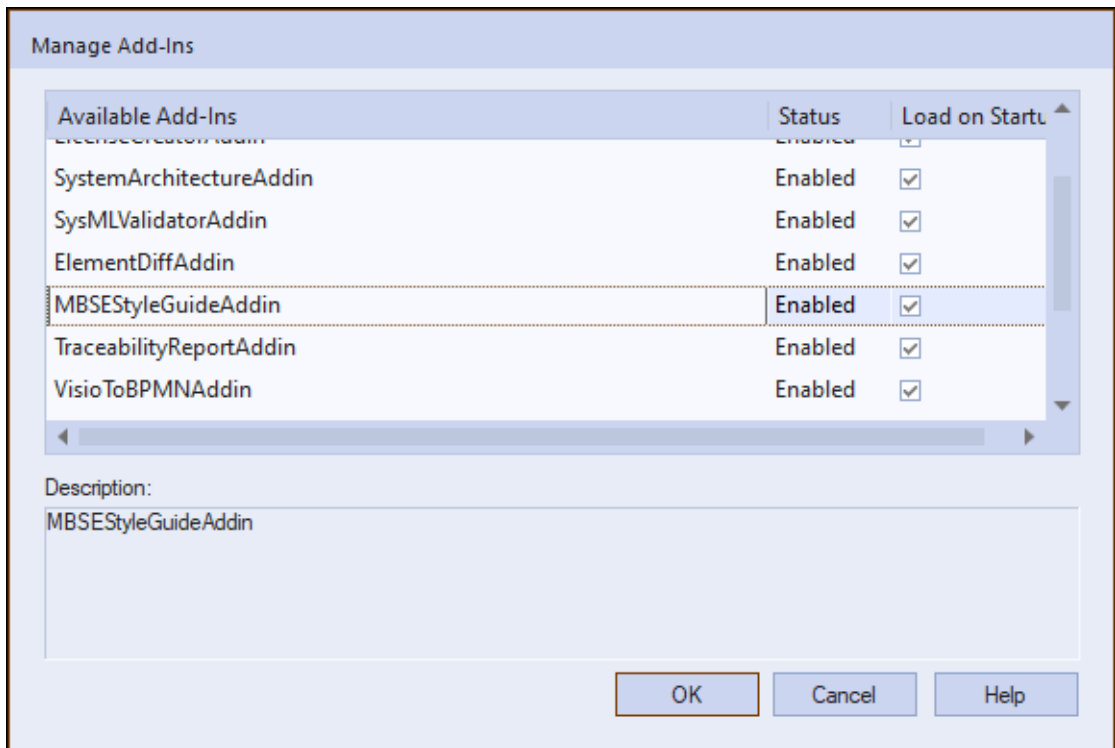
Bring up Enterprise Architect, without necessarily opening a repository, and verify that the *MBSE Style Guide Validator* extension has been loaded using the *Specialize* → *Add-Ins* ribbon panel:



Should the extension **not** be present, use the same panel and select:



Then check if the *MBSE Style Guide Validator* extension is present and enabled:



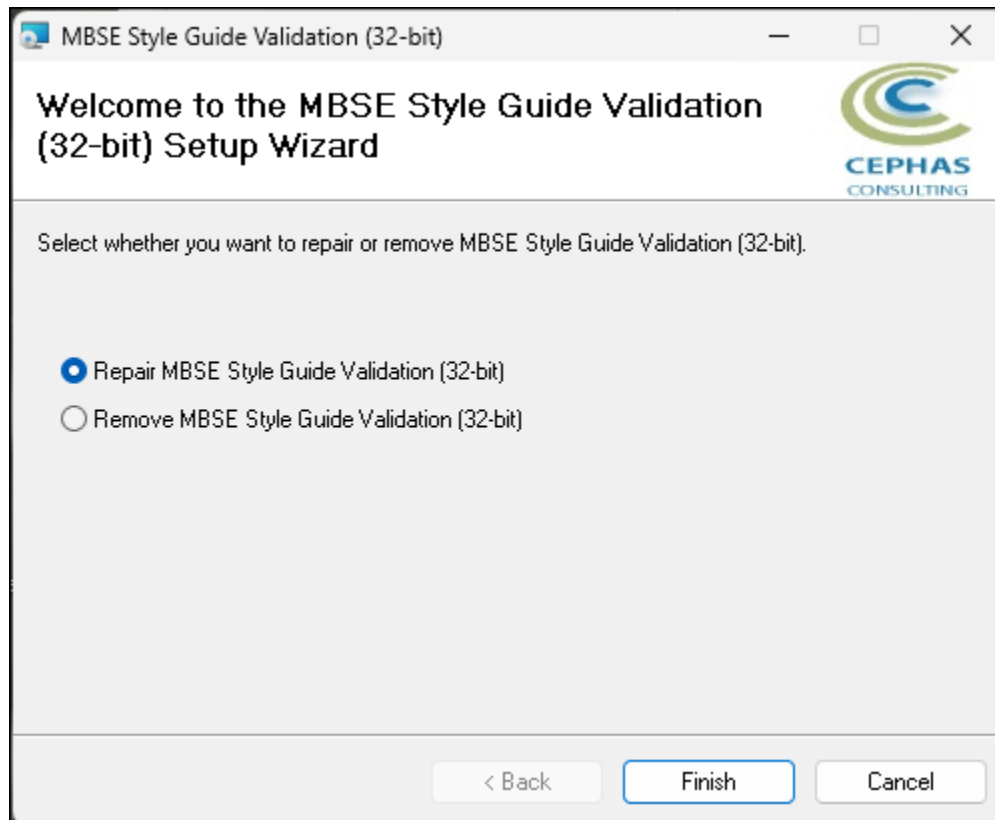
If an error status is shown, this typically means that either:

- 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 *MBSE Style Guide Validator* entry itself is not found then the extension installation did not complete successfully.

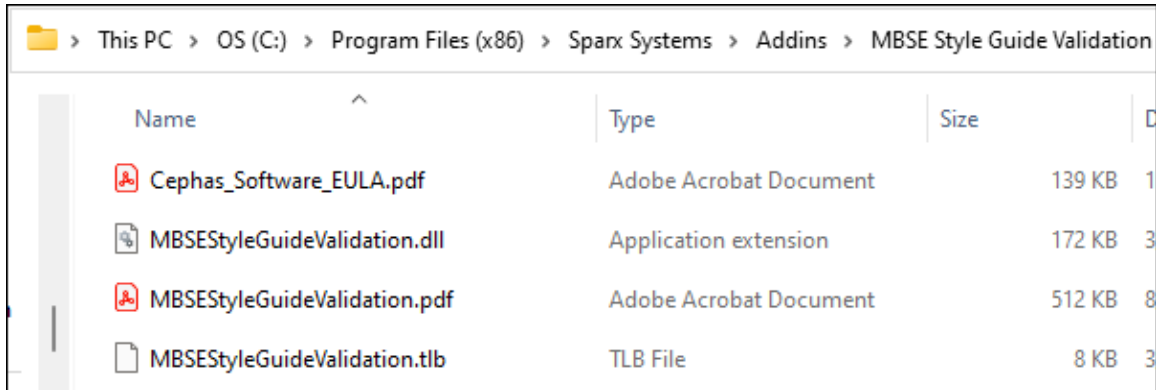
To fix an incorrect installation:

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



If, after the repair procedure, the 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, e.g. for the 32-bit version:



The screenshot shows a Windows File Explorer window with the following path: This PC > OS (C:) > Program Files (x86) > Sparx Systems > Addins > MBSE Style Guide Validation. The window displays a list of four files:

Name	Type	Size	Details
Cephas_Software_EULA.pdf	Adobe Acrobat Document	139 KB	1
MBSEStyleGuideValidation.dll	Application extension	172 KB	3
MBSEStyleGuideValidation.pdf	Adobe Acrobat Document	512 KB	8
MBSEStyleGuideValidation.tlb	TLB File	8 KB	3

Licensing

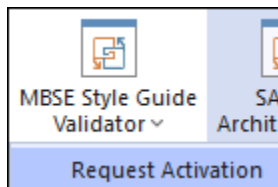
Trial version

The software installation automatically loads the trial version license key. No further action is required on the part of the end User.

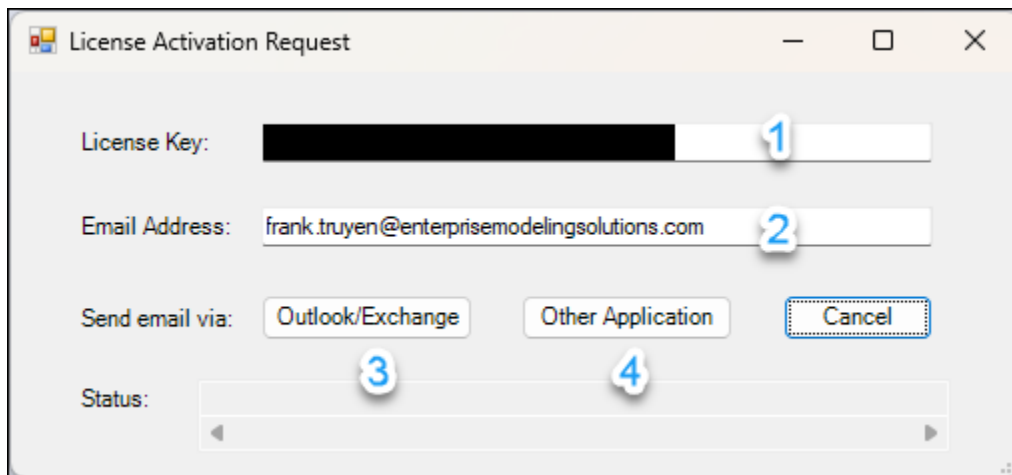
Licensed version

Once the full version of the product has been [purchased](#), a license key will be provided by Cephass Consulting via email. Once that key has been received, proceed with the following steps:

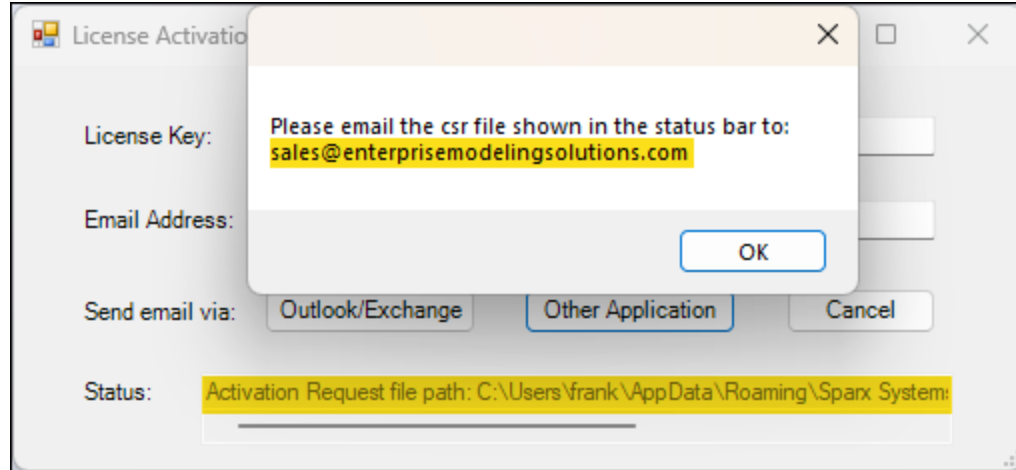
Request the license activation



Note: the license, once activated, is associated with your Windows login User ID (without the network domain information).



1. Paste in the license key from the email you received.
2. Provide the email address where you wish to receive the license activation notification.
3. Click this option if you use Microsoft Outlook with Exchange as your email application. This automatically emails your activation request (CSR file) to Cephass Consulting.
4. Otherwise, click this option to send the email yourself. The file path to the generated activation request (CSR file) is displayed in the Status bar of the dialog:



Attach the CSR file to your email and send the request to:

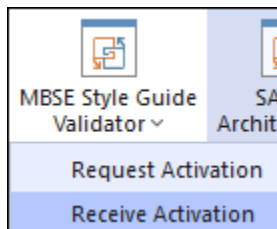
sales@enterprisemodelingsolutions.com

Receive the license activation

Upon receiving your activation request, Cephass Consulting replies with a license key file named "Fixed.keys".

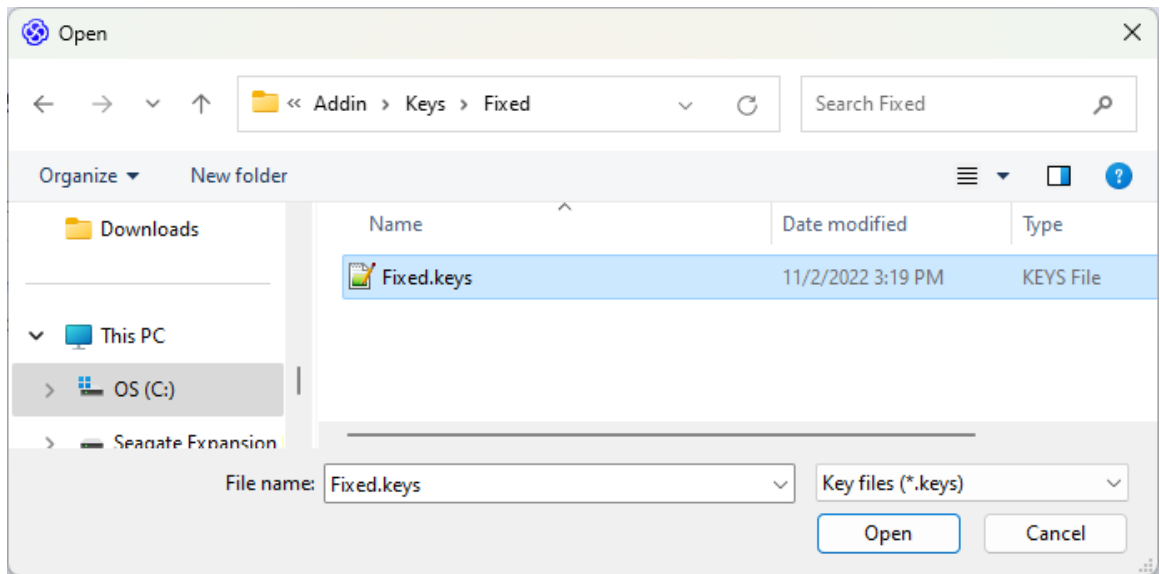
Caveat: this file can be saved anywhere (e.g. for backup purposes) on your system, but its name should not be changed.

To install the license key file, open Enterprise Architect and in the *Specialize* → *Add-Ins* ribbon panel, select:

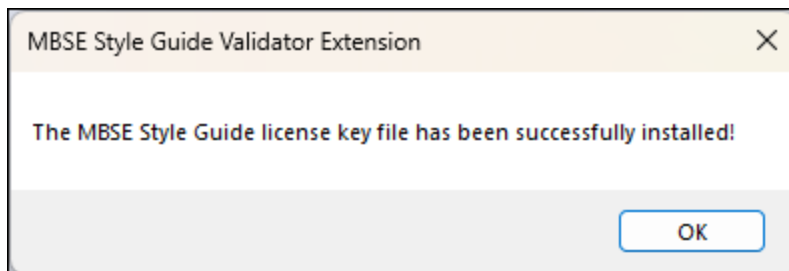


MBSE Style Guide

Next, select the file you copied out of the received email. For example:



A confirmation of the license activation is then displayed:



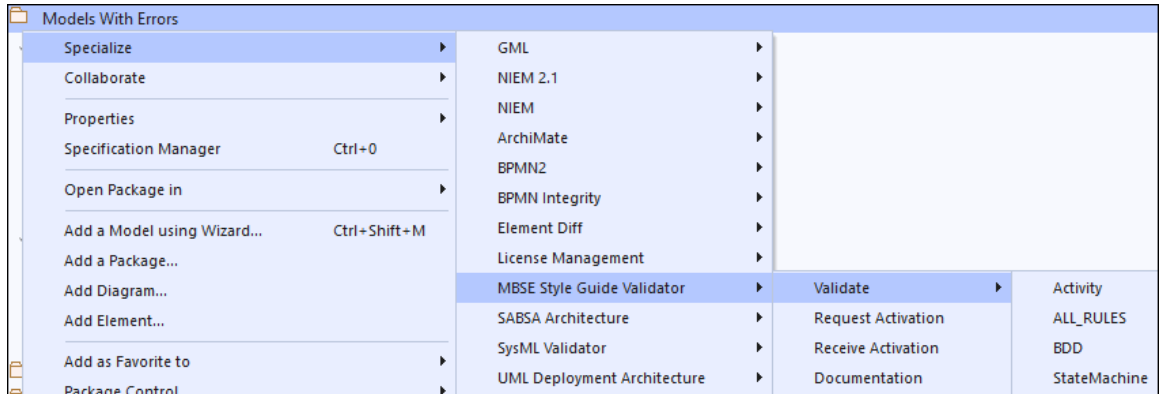
Next **exit the Enterprise Architect tool** and **restart it** in order to fully enable the extension.

Running the Style Guide validation

Package level validation

Note: 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).

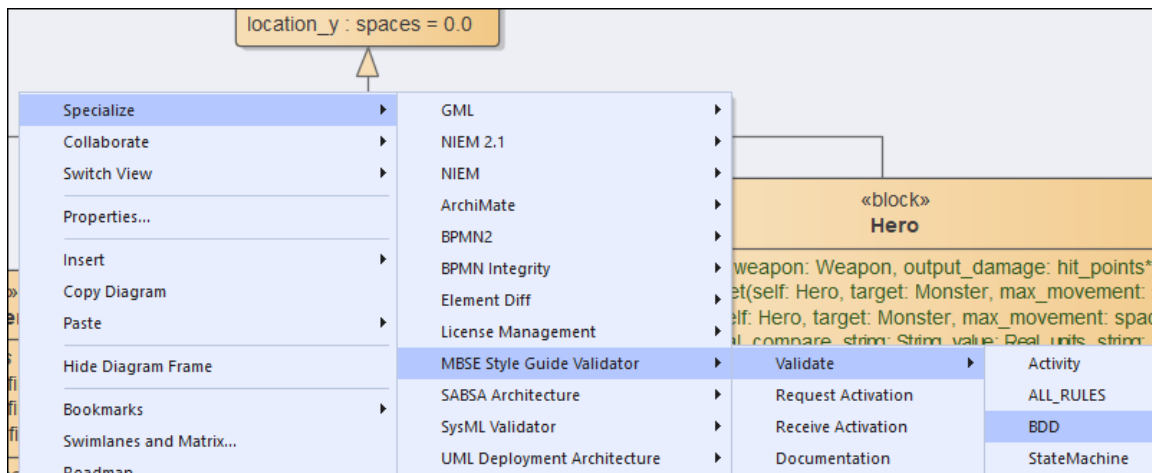
Right click a Package, or the top of a Package hierarchy, and select any of the available [rule sets](#):



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

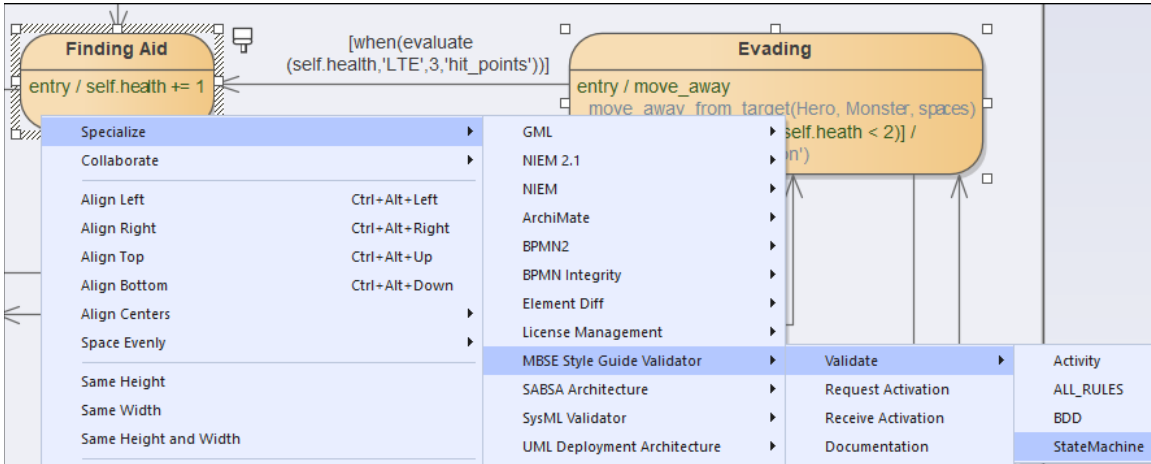
Diagram level validation

Right click a diagram background, and select the rule set appropriate for this type of diagram. For example, in the context of a SysML BDD:



Element level validation

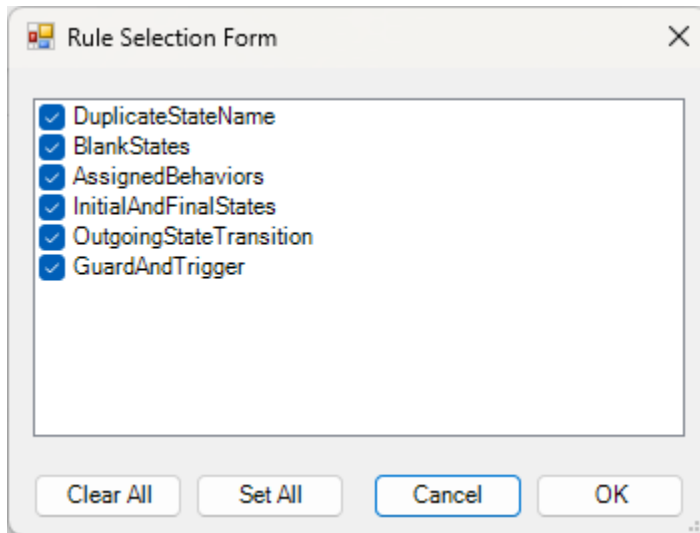
Select one or more diagram elements, then **right click** and select the rule set appropriate for this type of selection. For example, in the context of a SysML State Machine diagram:



If the set contains rules for validating connectors, ensure that both the source and the target elements are part of your selection.

User Interface

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



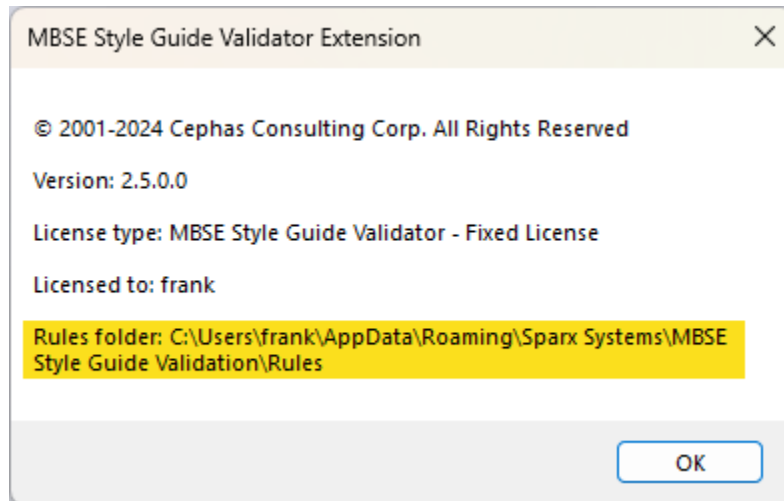
This allows rules to be enabled/disabled for a specific validation. To disable rules by default, create a custom rule set (see next section).

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](#). For example:



- 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.

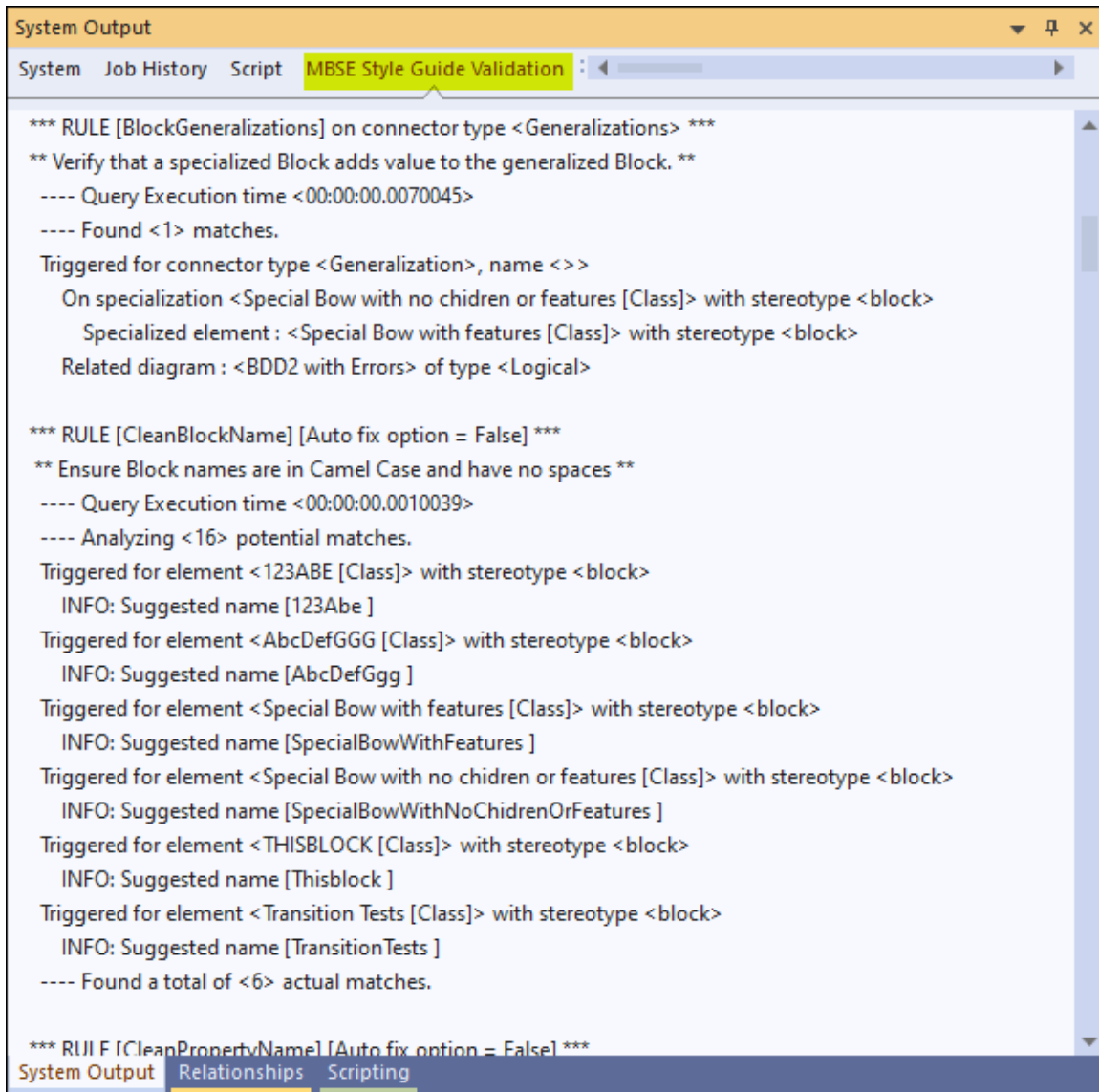
When adding a new rule to a set, ensure that it is given a unique ID (i.e., name) value.

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 “MBSE Style Guide Validation” tab in the System Output window automatically opens and displays the results of the validation. For example:



```

System Output
System Job History Script MBSE Style Guide Validation
*** RULE [BlockGeneralizations] on connector type <Generalizations> ***
** Verify that a specialized Block adds value to the generalized Block. **
---- Query Execution time <00:00:00.0070045>
---- Found <1> matches.
Triggered for connector type <Generalization>, name <>>
  On specialization <Special Bow with no children or features [Class]> with stereotype <block>
    Specialized element : <Special Bow with features [Class]> with stereotype <block>
    Related diagram : <BDD2 with Errors> of type <Logical>

*** RULE [CleanBlockName] [Auto fix option = False] ***
** Ensure Block names are in Camel Case and have no spaces **
---- Query Execution time <00:00:00.0010039>
---- Analyzing <16> potential matches.
Triggered for element <123ABE [Class]> with stereotype <block>
  INFO: Suggested name [123Abe ]
Triggered for element <AbcDefGGG [Class]> with stereotype <block>
  INFO: Suggested name [AbcDefGgg ]
Triggered for element <Special Bow with features [Class]> with stereotype <block>
  INFO: Suggested name [SpecialBowWithFeatures ]
Triggered for element <Special Bow with no children or features [Class]> with stereotype <block>
  INFO: Suggested name [SpecialBowWithNoChildrenOrFeatures ]
Triggered for element <THISBLOCK [Class]> with stereotype <block>
  INFO: Suggested name [Thisblock ]
Triggered for element <Transition Tests [Class]> with stereotype <block>
  INFO: Suggested name [TransitionTests ]
---- Found a total of <6> actual matches.

*** RULE [CleanPropertyName] [Auto fix option = False] ***
System Output Relationships Scripting
  
```

Every rule match listed in this window uses 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.

In other cases, additional output lines may be added to include related diagrams, object Classifiers, etc.

Active output lines are the ones that are NOT:

- blank
- starting with the characters '*' or '-'.
- starting with the text 'INFO:'

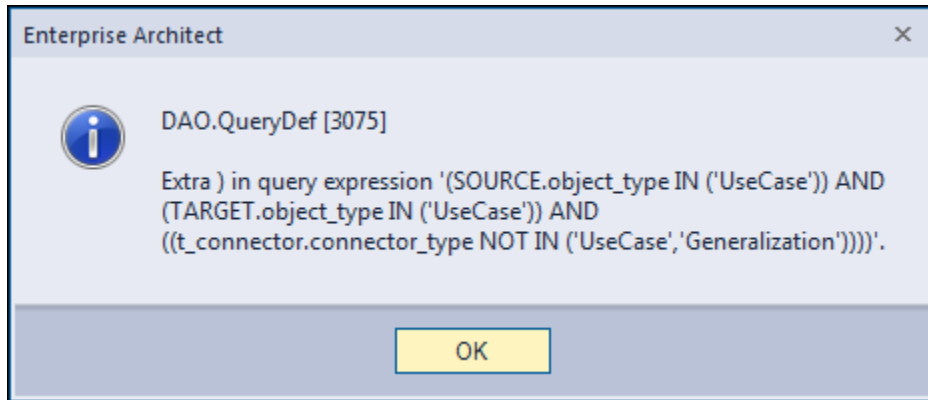
Single click an active output line to automatically locate its related element, Attribute, Operation or diagram in the Browser.

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

For connector related rules, single or double click attempts to open up at least one diagram where the connector is graphically represented (if any), and selects the connector in that diagram (you may need to scroll the window to see the selection).

Troubleshooting

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



As of version 16.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:

```
*** RULE [BlockGeneralizations] on connector type <Generalizations> ***
```

- If you are validating the repository using a [customized rule set](#), please include that xml file in the data provided back to Cephias.
- Also provide:
 - Your database type (SQLite, SQL Server, Oracle, etc.) and version number.
 - The version of the MBSE Style Guide 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: contact@enterprisemodelingsolutions.com

Phone : 208-462-4863.