

BPMN Integrity Extension

BPMN Integrity Extension	1
Disclaimer	4
Dependencies	4
Limitations of the trial version	4
Installation	4
Verifying the installation.....	6
Installing the license key file	10
Adding the User license key	12
Auto-numbering of BPMN process models.....	16
Overview	16
Performing the auto-numbering	17
Sample Auto-Numbered Process	20
Displaying the number values in the diagram	20
Removing the automatically assigned numbers.....	21
BPMN process integrity validation	22
Overview	22
Limitations of the current implementation	22
Performing the process integrity validation	23
Default Rule Set	29
Customizing the Rule Set	30
Validation Rules	32
Rule BIR_01	32
Rule BIR_02	32
Rule BIR_03	33
Rule BIR_04	33
Rule BIR_05	34
Rule BIR_06	34
Rule BIR_07	35
Rule BIR_08	35
Rule BIR_09	35

Rule BIR_10.....	36
Rule BIR_11.....	36
Rule BIR_12.....	36
Rule BIR_13.....	36
Rule BIR_14.....	37
Rule BIR_15.....	37
Rule BIR_16.....	37
Rule BIR_17.....	38
Rule BIR_18.....	38
Rule BIR_19.....	38
Rule BIR_20.....	39
Rule BIR_21.....	40
Rule BIR_22.....	40
Rule BIR_23.....	41
Rule BIR_24.....	41
Rule BIR_25.....	41
Rule BIR_26.....	42
Rule BIR_27.....	42
Rule BIR_28.....	42
Rule BIR_29.....	43
Rule BIR_30.....	43
Rule BIR_31.....	43
Rule BIR_32.....	44
Rule BIR_33.....	44
Rule BIR_34.....	44
Rule BIR_35.....	45
Rule BIR_36.....	45
Rule BIR_37.....	45
Rule BIR_38.....	45
Rule BIR_39.....	46
Rule BIR_40.....	46
Rule BIR_41.....	47
Rule BIR_42.....	47



BPMN Integrity User Guide

Rule BIR_43.....	47
Rule BIR_44.....	48
Rule BIR_45.....	48
Rule BIR_46.....	48
Rule BIR_47.....	49
Troubleshooting	50
Support and contact information.....	50

Disclaimer

The guidelines contained in this document are based on release 12.1 and 13.x of Enterprise Architect (EA). Version 1.1 of the *BPMN Integrity Extension* has been successfully tested for deployment with both these versions.

This deployment, as well as the 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 12.1 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 4.0.

Limitations of the trial version.

The following limitations apply to the trial version:

- The software activation is granted for five (5) consecutive days only.
- Only the first fifty (50) [integrity rule](#) violations are reported back to the User.
- Rule disabling and customization is not supported (i.e. all rules will be executed).
- Automatic fixing of detected violations is disabled.

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.

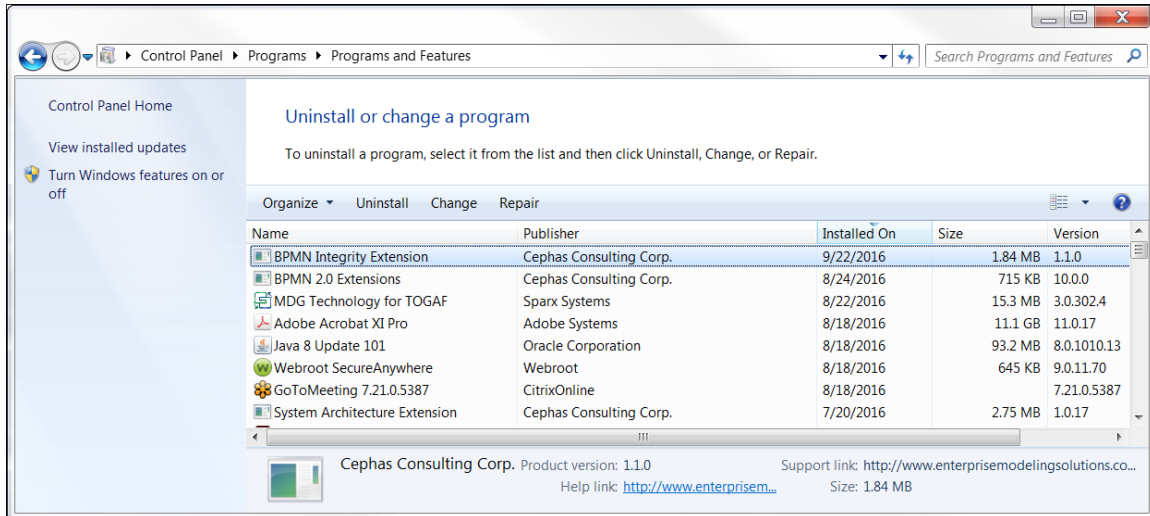
The installation will attempt to update the Windows registry, so the User needs to ensure that s/he has sufficient privileges to run the setup program.

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.

Software upgrades

Older versions of the software are automatically removed and replaced. However, if problems are encountered during an upgrade, the older version can be removed manually using the *Control Panel* → *Uninstall or change program* interface. In that case, look for the program name “BPMN Integrity Extension”.

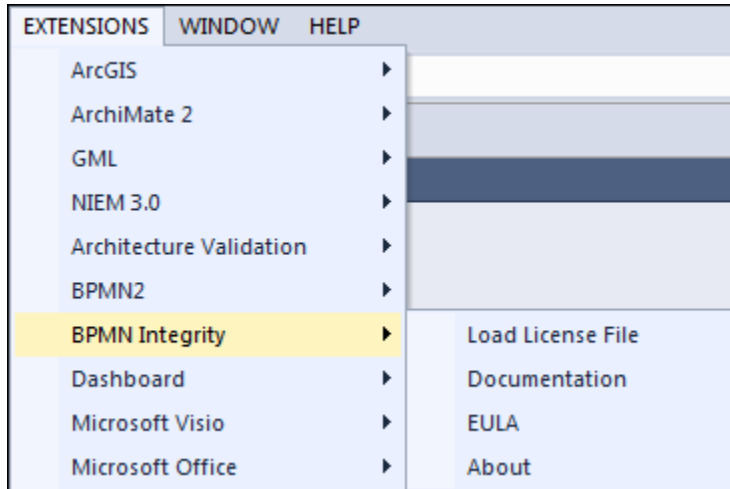


Should the installation fail for any reason other than insufficient User privileges, please take appropriate screenshots and email the data to the [support](#) address listed at the bottom of this document.

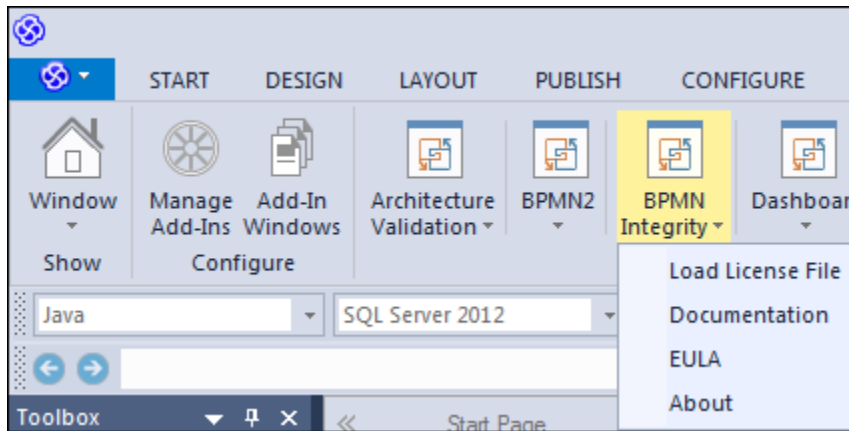
Verifying the installation

Bring up Enterprise Architect, without necessarily opening a repository, and verify that there is a *BPMN Integrity* entry.

In version 12.1, under the EXTENSIONS menu:



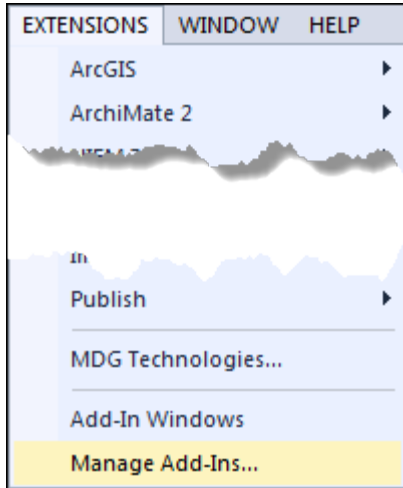
In version 13.x, in the EXTEND ribbon:



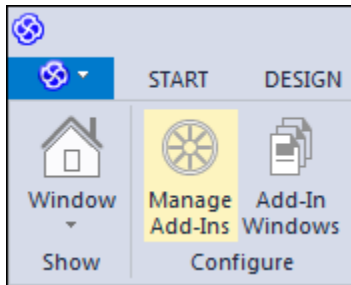
Note that the other extensions shown in the above screenshots may or may not be present, depending on your Enterprise Architect version and configuration.

Should the menu entry not be present,

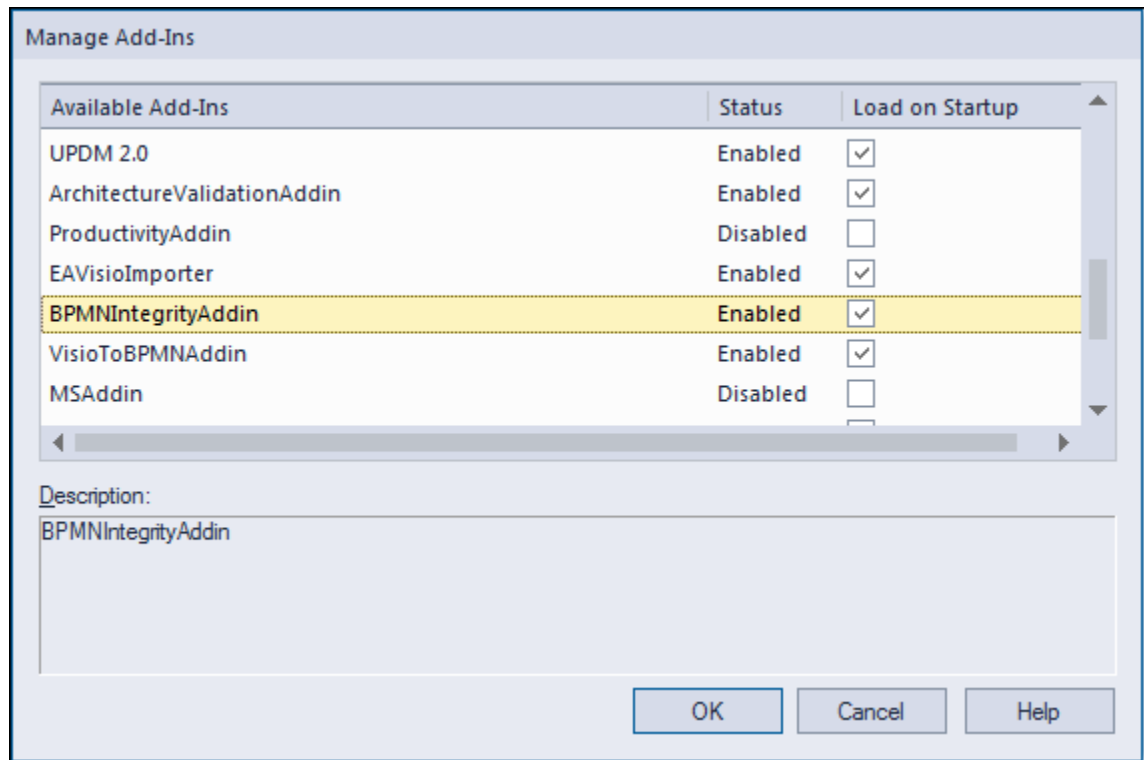
In version 12.1 select the “EXTENSIONS → Manage Add-Ins” menu option:



In version 13.x, select “Manage Add-Ins” in the EXTEND ribbon:



And confirm that the *BPMN Integrity* extension is loaded and enabled:



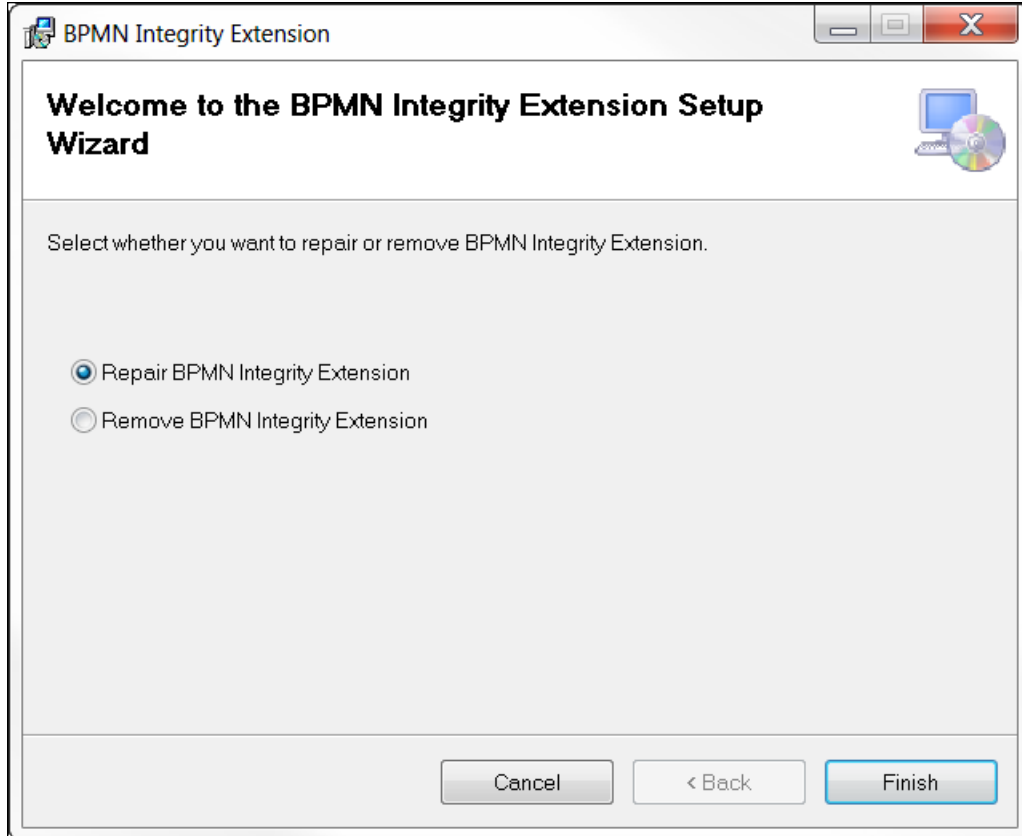
If an error status is shown this typically means that either:

- 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 *BPMN Integrity* 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 *BPMN Integrity* 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
 BPMNIntegrityAddin.dll	Application extens...	161 KB
 BPMNIntegrityAddin.tlb	TLB File	2 KB
 BPMNIntegrityExtension.pdf	Adobe Acrobat D...	856 KB
 Cephass_Software_EULA.pdf	Adobe Acrobat D...	60 KB
 Cephass_Software_EULA.rtf	Rich Text Format	126 KB
 cephas-logo-h-banner.jpg	JPEG image	7 KB
 EA.TLB	TLB File	217 KB
 Interop.EA.dll	Application extens...	300 KB
 register_BPMNIntegrityAddin.bat	Windows Batch File	1 KB
 Unregister_BPMNIntegrityAddin.bat	Windows Batch File	1 KB

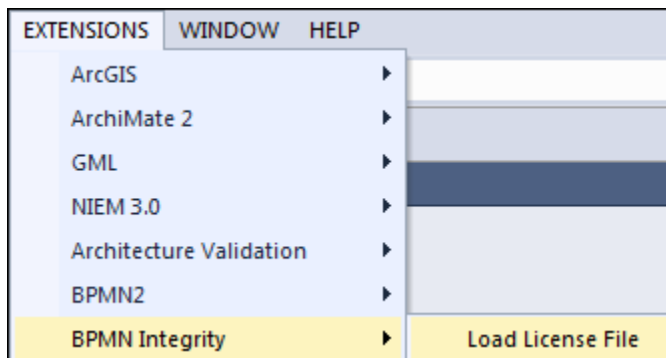
Installing the license key file

By default the installation loads the trial version license key.

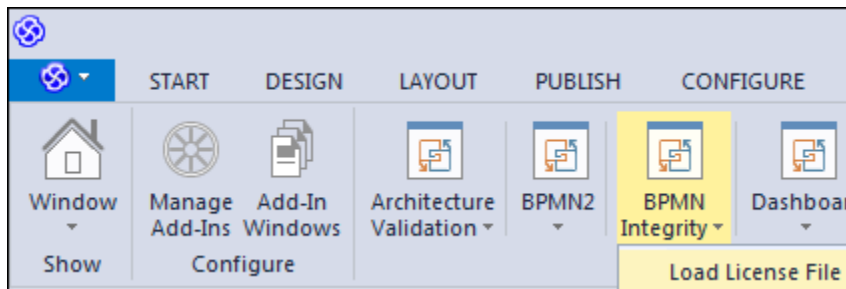
Once the full version of the product has been purchased, a *Fixed.keys* (or *Shared.keys*) file will be provided by Cephass Consulting which needs to be installed **by each licensed User of the software**, even if floating license keys are purchased.

To install the license key file, open Enterprise Architect and click:

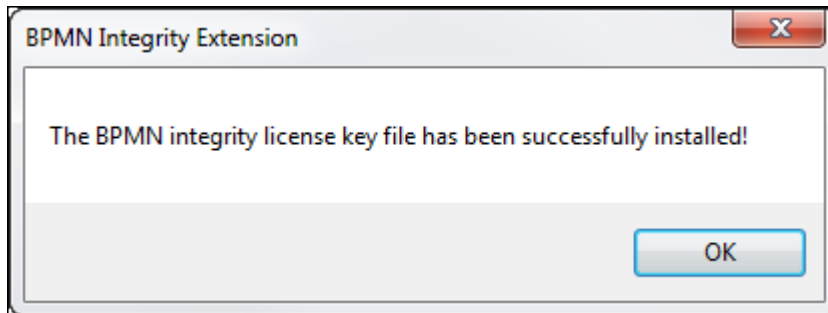
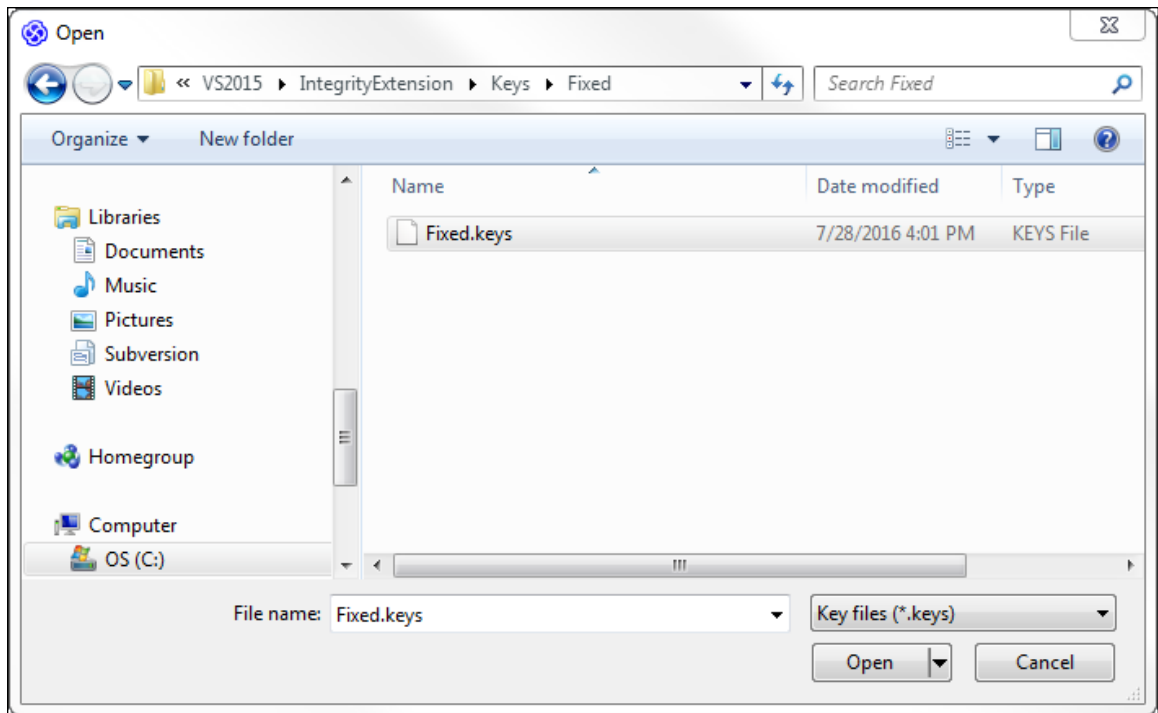
In version 12.1:



In version 13.x:



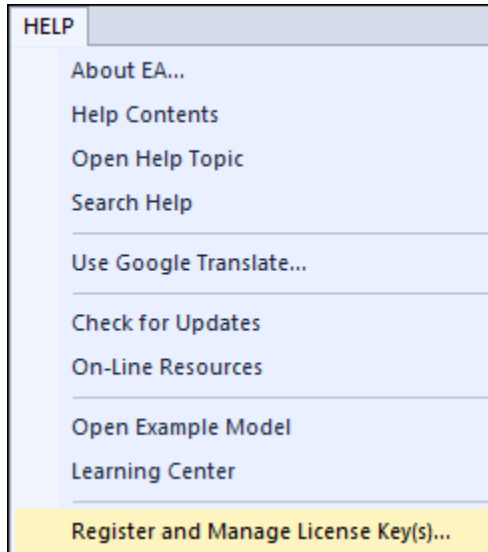
Next, select the provided file. For example:



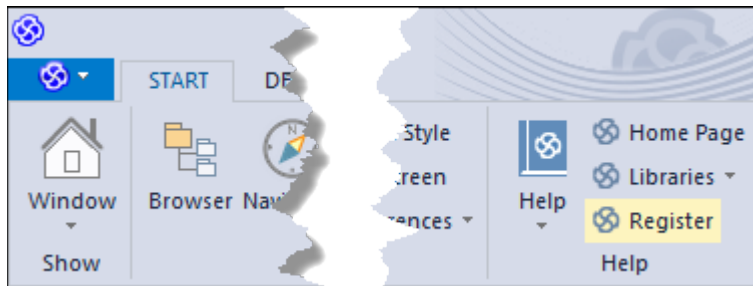
Adding the User license key

After installing the license key file, the following step is required for both the trial and the full version of the software.

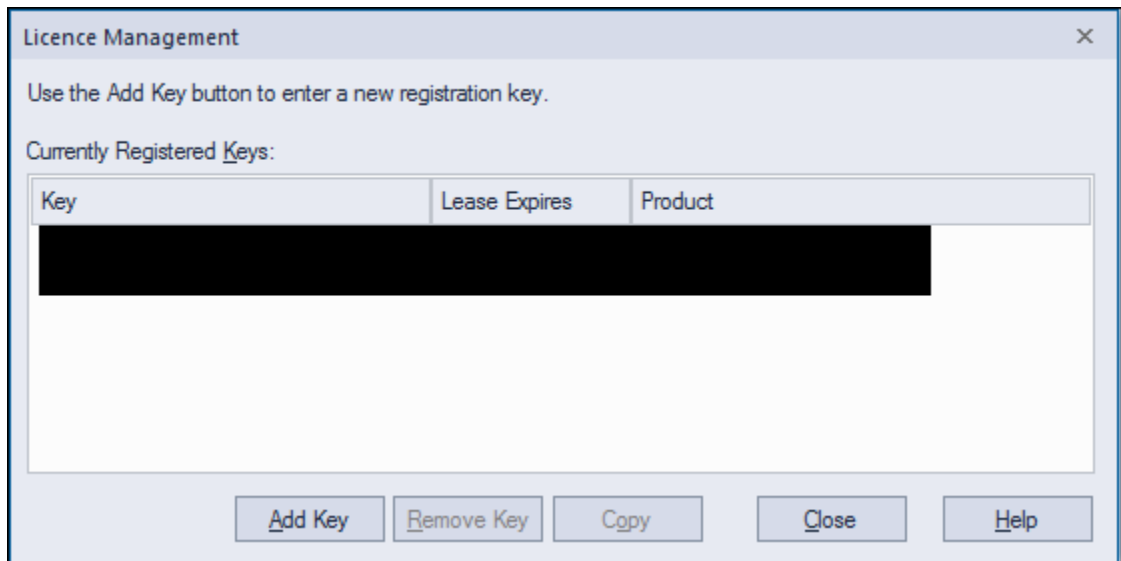
In version 12.1, under the HELP menu, select:



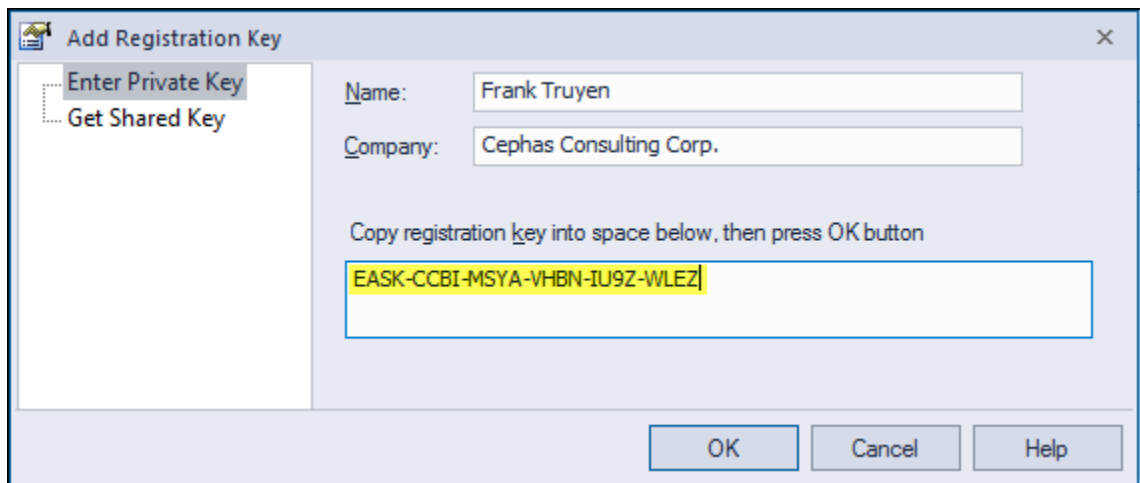
In version 13.x, in the START ribbon, select:



Next, click “Add Key”:

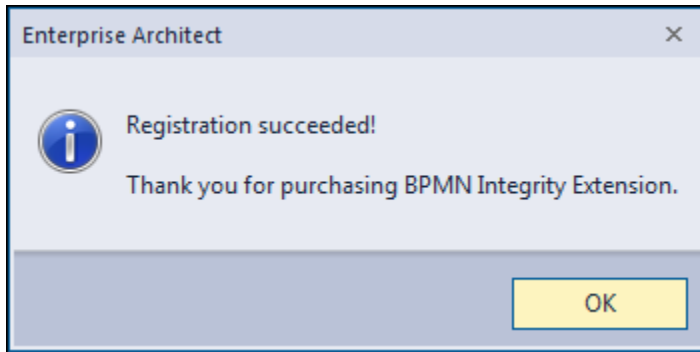


For **fixed (private) licenses**, enter or copy/paste, either the trial key (shown below), or one the full version keys provided as part of the software purchase:

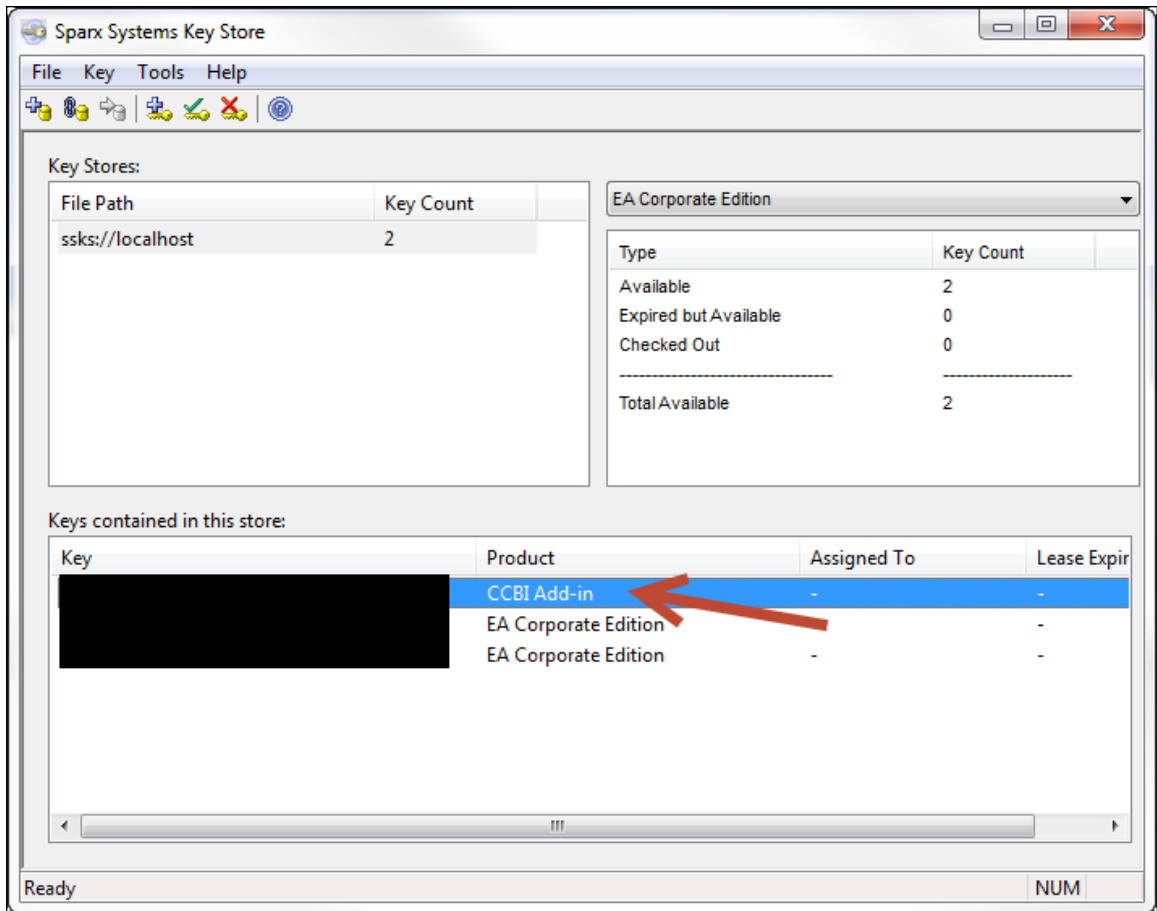


Trial key: EASK-CCBI-MSYA-VHBN-IU9Z-WLEZ

Enterprise Architect will confirm the successful addition of a key:

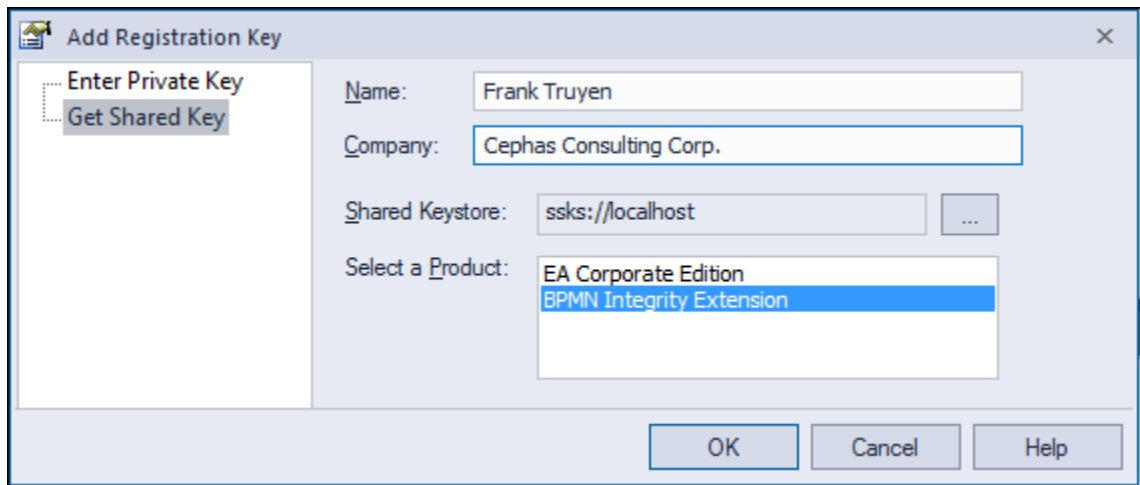


For **floating licenses**, first the administrator needs to add the key/s to the Sparx System key store (**version 2.3 or higher**), using the same process as for Enterprise Architect license keys:



BPMN Integrity User Guide

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



The screenshot shows a dialog box titled "Add Registration Key" with a close button (X) in the top right corner. On the left side, there are two tabs: "Enter Private Key" and "Get Shared Key", with the latter being selected. The main area contains the following fields and controls:

- Name:** A text box containing "Frank Truyen".
- Company:** A text box containing "Cephas Consulting Corp.".
- Shared Keystore:** A text box containing "ssk://localhost" and a browse button (three dots).
- Select a Product:** A list box containing "EA Corporate Edition" and "BPMN Integrity Extension", with the latter selected and highlighted in blue.

At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Auto-numbering of BPMN process models

Overview

This feature automatically numbers the steps (Activities only) of a BPMN 2.0 business process. Each step is numbered using the actual process flow as its sequencing mechanism and with an increment of 10 (e.g. 100.010, 100.020, 100.030, etc.). This allows for up to 9 steps to be easily inserted without requiring renumbering, while keeping a consistent numbering scheme.

Sub-process steps are automatically numbered based on their parent number (e.g. 100.020.010, 100.020.020, 100.020.030, etc. where 100.020 is the number of the sub-process itself).

To best leverage the tool features, the number is stored in the Alias field, independently of the Name field. Name and/or Alias fields can be shown in the diagram. The Alias field is not displayed by default – this can be enabled on a per diagram basis via the diagram properties dialog (see [screenshot](#)).

Note that events (Start, End and Intermediate) and gateways are not numbered.

Numbering can start from:

1. Any sequenced element (typically a Start Event or the first Activity) of the process flow (at the top level, or at the sub-process level). Any process step preceding the selected element is unaffected, unless it is the target of a downstream Sequence Flow (e.g. in a loop flow).
2. The top level Business Process element. The extension will look for a child diagram under the Process and search that diagram for any elements that start a flow (typically a Start Event or an Activity with no incoming Sequence Flow) and then number each flow as an independent thread.

The Business Process element is assigned the starting number (e.g. 100), or the prefix if the starting number provided by the User contains more than one segment (e.g. if the starting number given is 100.10, then the Business Process element is assigned the prefix 100, to document that all its process steps will start with that number).

3. A BPMN 2.0 diagram. The extension will search that diagram for any elements that start a flow (typically a Start Event or an Activity with no incoming sequence flow) and then number each flow as an independent thread.

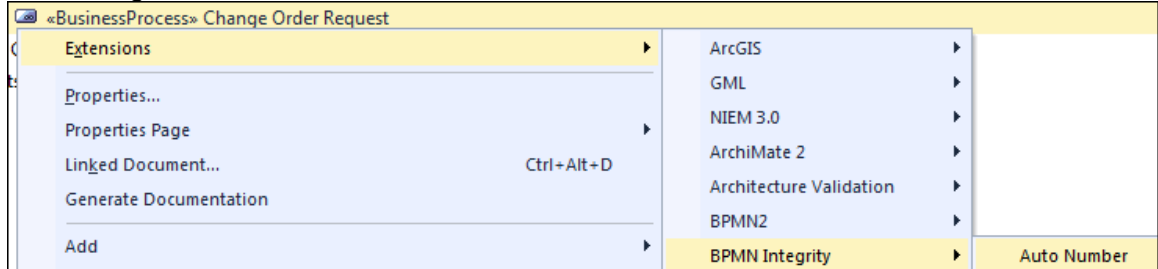
In either case, the User will be given the option to either keep the current number (if the starting element already had a number assigned to the Alias field) or assign a new starting number.

An Activity will only be assigned a (new) number once, the first time it is encountered in the flow.

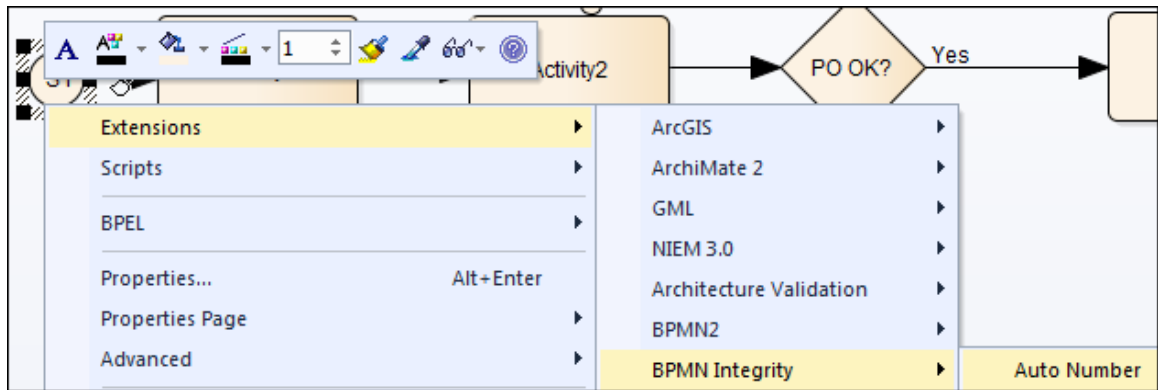
For Sequence Flows coming out of Gateways, a flow with either “yes” or “true” in the name field (or in the *conditionExpression* tagged value) will be numbered before any of the other outgoing named or unnamed flows.

Performing the auto-numbering

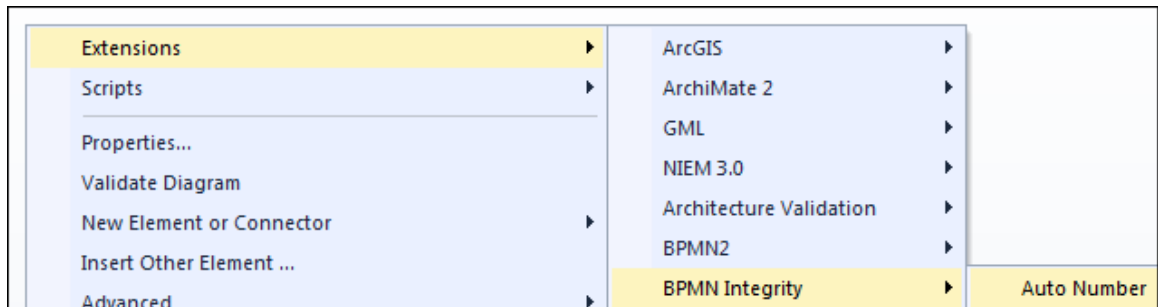
1. In the Project Browser: right click the Business Process element as the numbering context:



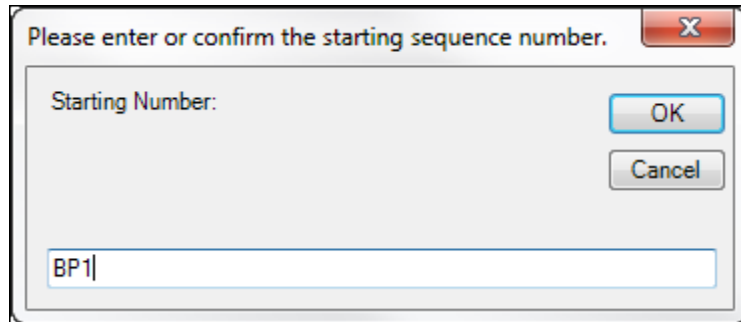
2. In a BPMN 2.0 diagram, right click a Business Process, Start Event, Gateway or Activity as the starting element:



3. Or right click the BPMN 2.0 diagram background:



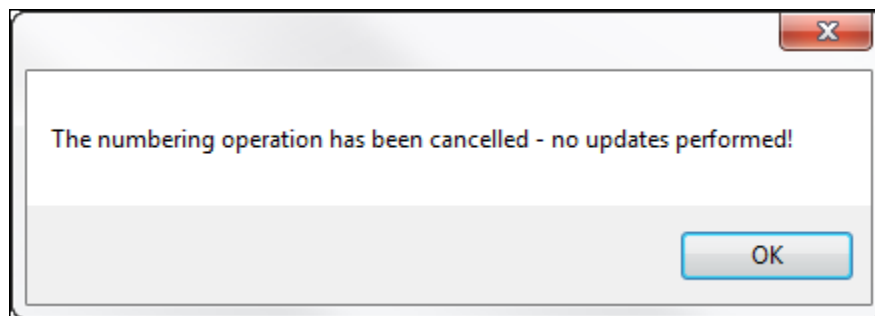
Once the program has located at least one element that starts the flow, the following dialog is presented:



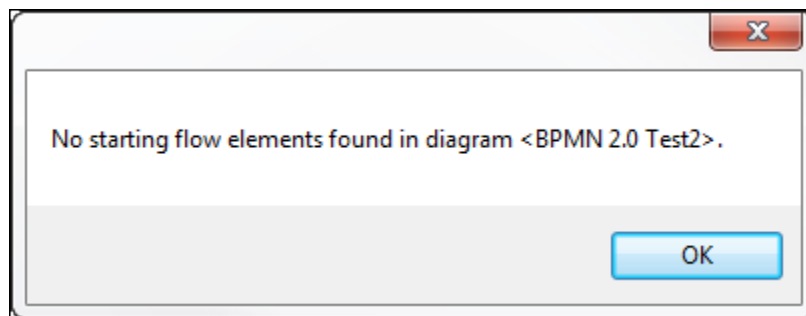
Note that, as in this example, the number prefix does not have to be numeric: the program automatically appends the starting numeric value (e.g. "BP1.010"). It is recommended to keep the prefix string/number small since every additional sub-process level will expand the complete number with another 4 characters (e.g. "BP1.150.060.040.010" at sub-process level 3).

To start at a specific number provide it in the text box (examples: "BP1.050", "100.040"). All numeric values of less than 3 digits in length are automatically formatted for consistency and alphabetic sorting.

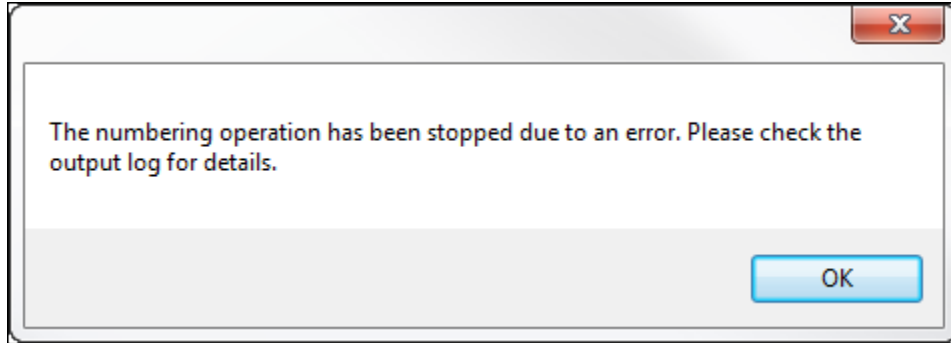
If "Cancel" is selected the numbering operation is aborted:



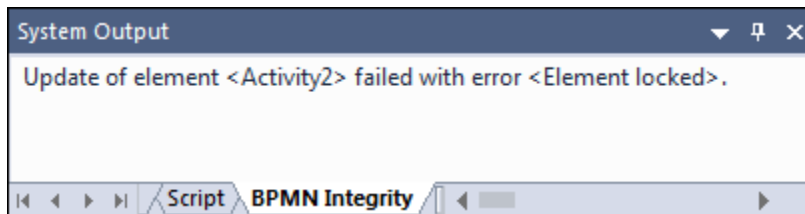
If no starting flow elements can be found (e.g. the diagram is empty, or the elements on the diagram have not been connected with Sequence Flows) this error message will be displayed:



If any element update fails (for example if the Package containing the process model is locked), the numbering operation is immediately stopped and the following error message is displayed:

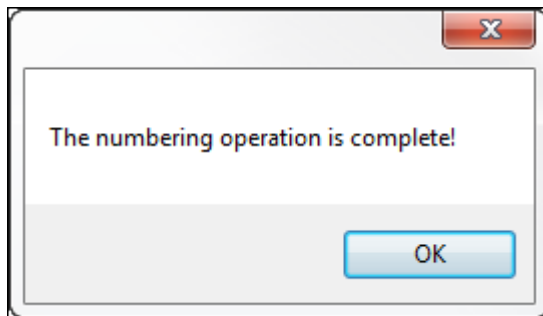


The program automatically opens the System Output window and creates a "BPMN Integrity" tab to log any issues encountered:

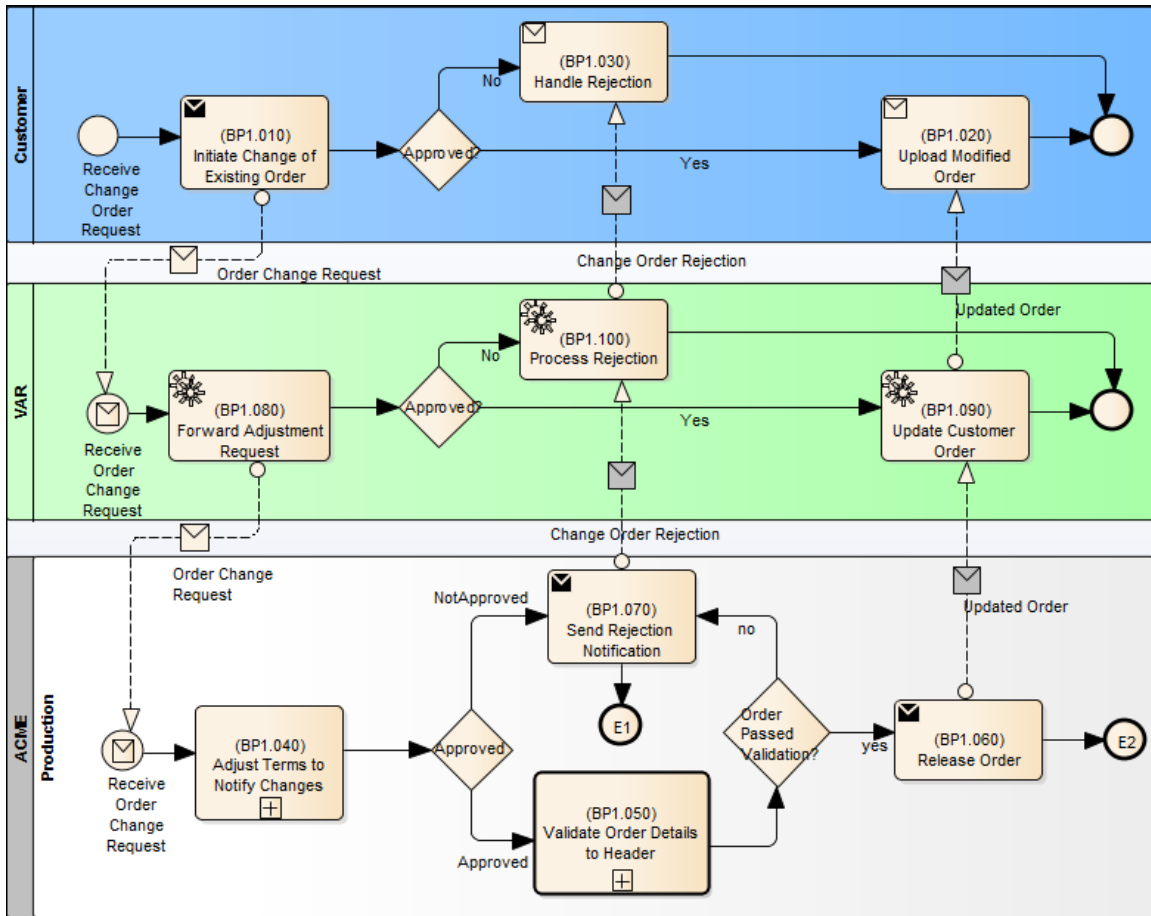


In this example the update failed because the parent package/folder of the process had not been checked out from version control.

The following message is displayed once the numbering procedure successfully completes:

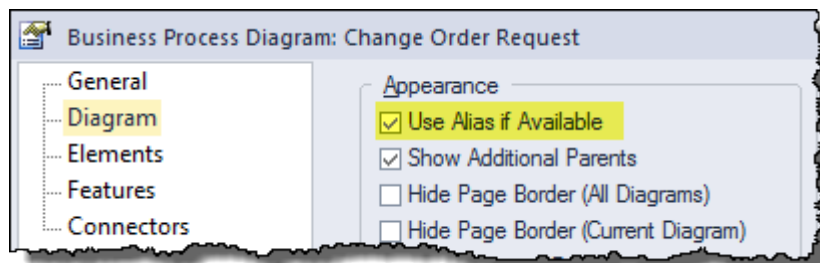


Sample Auto-Numbered Process



Displaying the number values in the diagram

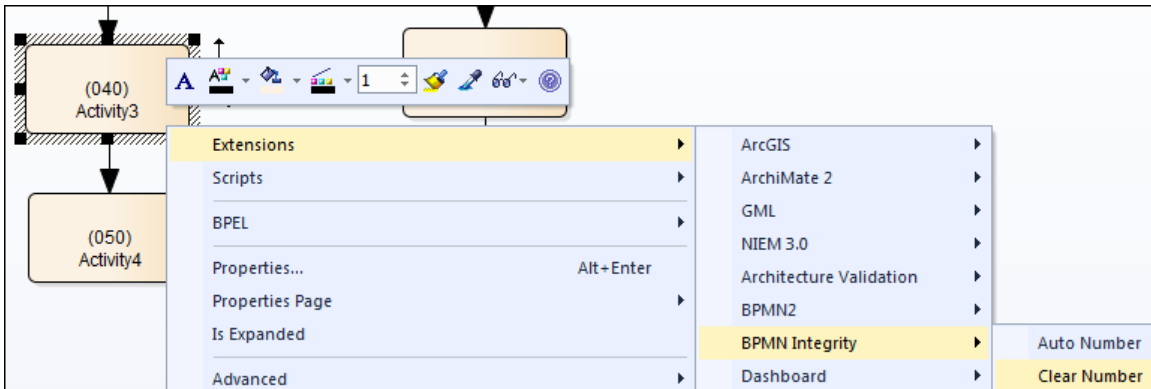
To make the numbers (stored in the Alias field) visible in the diagram, remember to enable the option in the diagram properties:



Removing the automatically assigned numbers

It is possible to also remove (i.e. clear out from the Alias field) previously assigned numbers using the “Clear Number” menu option. This can be accomplished from the same contexts as shown in [Performing the Auto Numbering](#).

For example, to remove the numbering starting from step 040 in the process model:



BPMN process integrity validation

Overview

This feature verifies the internal quality and coherence of process models based on a set of rules, some of which are based on the [BPMN Method and Style guidelines](#) developed by Bruce Silver.

Validation can occur at the following levels:

- A complete business process, including all sub-processes.
- A business process diagram (all diagram elements are validated, including sub-processes).
- One or more User selected elements on a diagram.

It is highly recommended that before running the Integrity validation the model is first verified for compliance with the BPMN 2.0 specification using the [free BPMN 2.0 syntax checker](#).

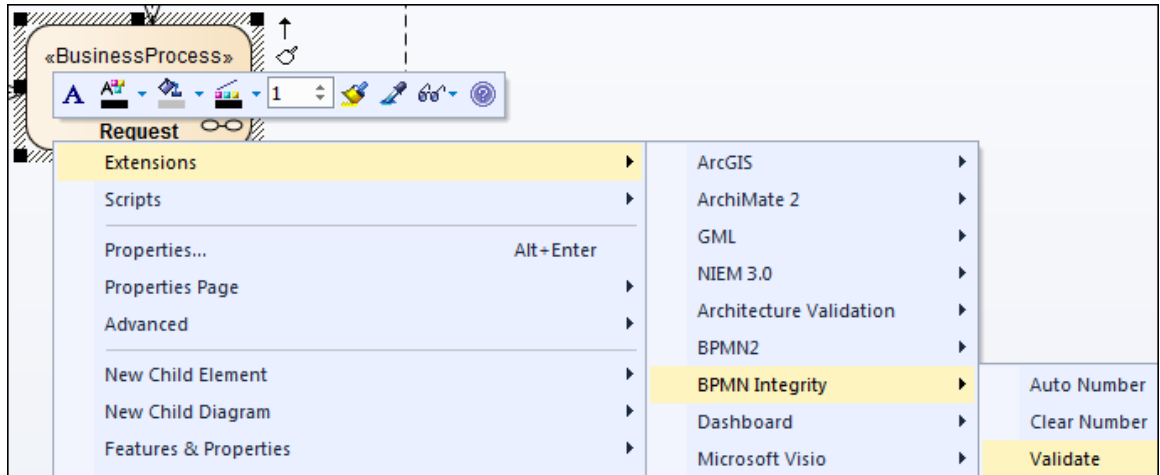
Limitations of the current implementation

- Only BPMN 2.0 Process and Collaboration models are validated, not Choreographies or Conversations.
- No explicit verification is performed for the following model constructs:
 - Compensation handling.
 - Start Events of type Multiple (they should be avoided whenever possible and replaced with discrete Start Events).
 - Gateways of type Complex (they should be avoided whenever possible since their simulation and/or execution can be problematic).

Performing the process integrity validation

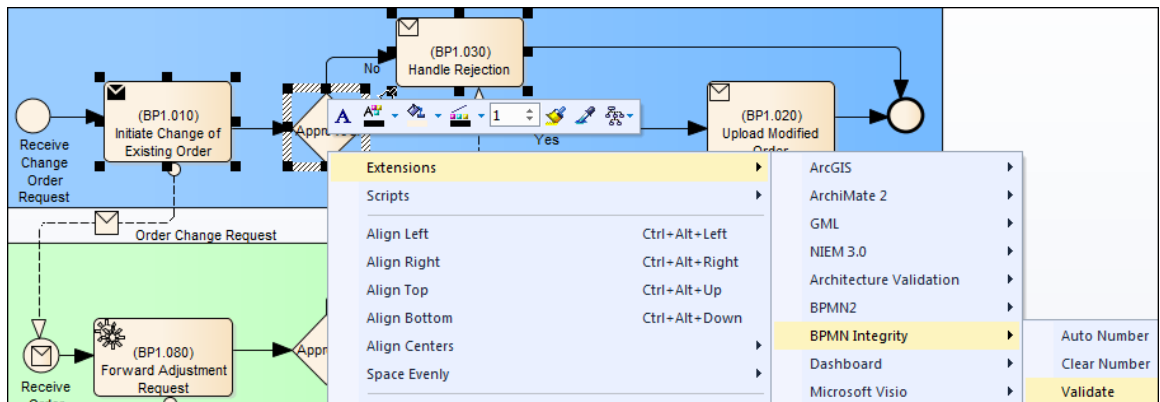
The verification can be started in different modeling contexts:

1. In a diagram: right click a Business Process element as the validation context:

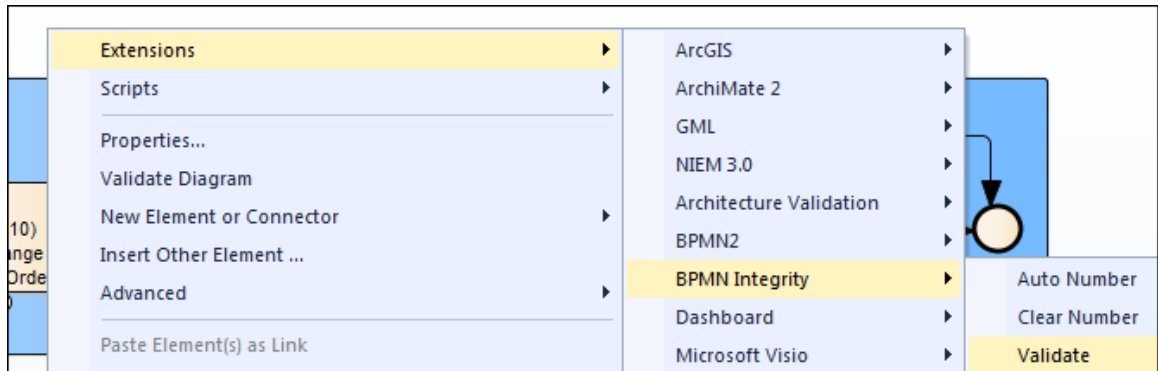


The program will traverse the process by validating all its child diagrams, including any sub-process levels.

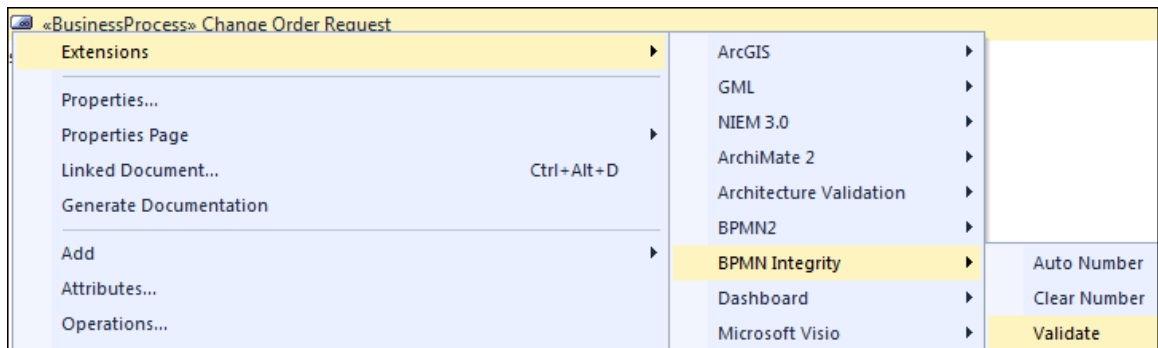
2. In a diagram, right click any one of a set of BPMN 2.0 selected elements as the context for the validation:



3. Or right click a BPMN 2.0 diagram background to validate all elements on the diagram. Sub-processes are automatically checked as well.

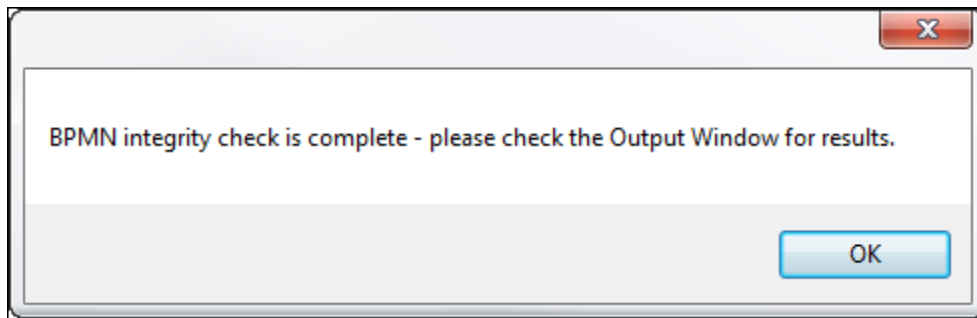


4. In the Project Browser, right click a Business Process element and select:

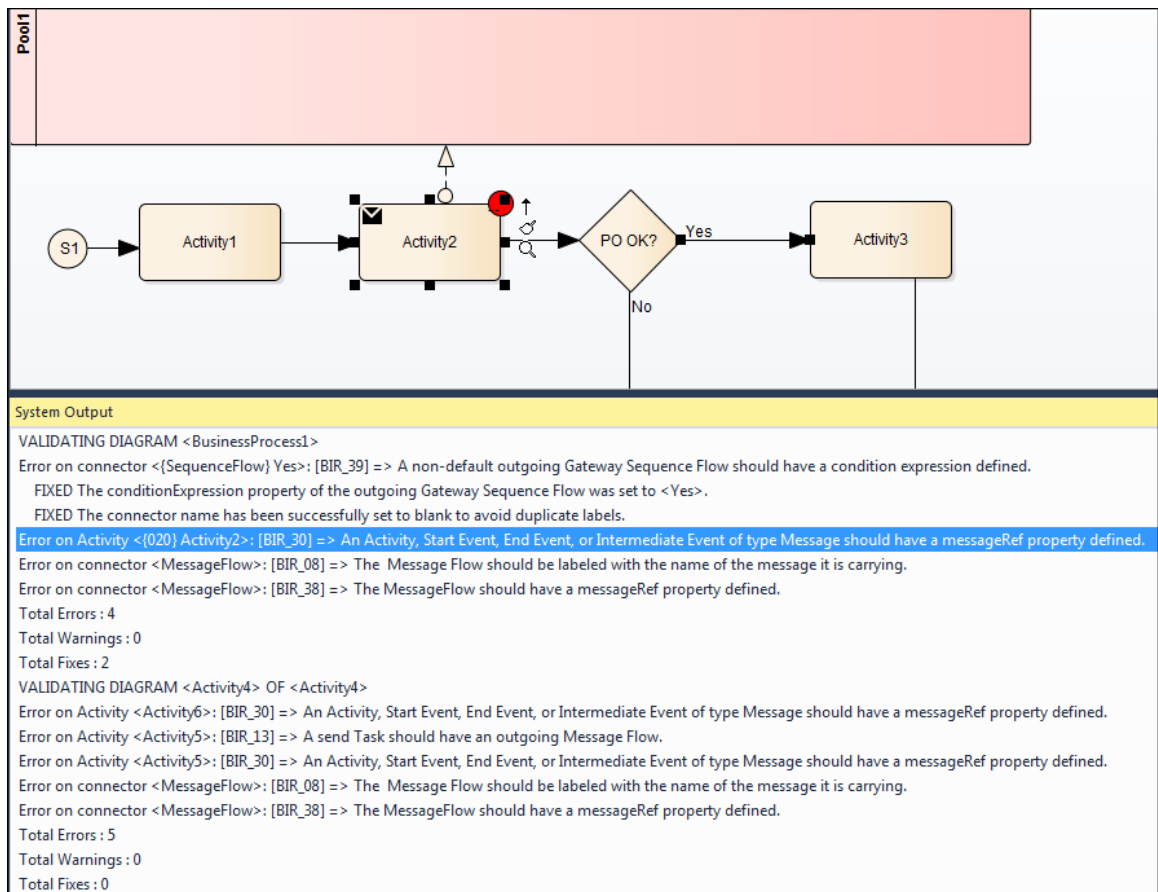


The program will traverse the process by validating all its child diagrams, including any sub-process levels.

Once the validation completes the following dialog is presented:



All validation errors are logged in the “BPMN Integrity” tab of the Enterprise Architect System Output window. This window is automatically opened by the program.



The screenshot shows a BPMN diagram with a start event (S1), Activity1, Activity2, a decision gateway (PO OK?), and Activity3. Activity2 has a red error icon. Below the diagram is the System Output window with the following log:

```

VALIDATING DIAGRAM <BusinessProcess1>
Error on connector <{SequenceFlow} Yes>: [BIR_39] => A non-default outgoing Gateway Sequence Flow should have a condition expression defined.
FIXED The conditionExpression property of the outgoing Gateway Sequence Flow was set to <Yes>.
FIXED The connector name has been successfully set to blank to avoid duplicate labels.
Error on Activity <{020} Activity2>: [BIR_30] => An Activity, Start Event, End Event, or Intermediate Event of type Message should have a messageRef property defined.
Error on connector <MessageFlow>: [BIR_08] => The Message Flow should be labeled with the name of the message it is carrying.
Error on connector <MessageFlow>: [BIR_38] => The MessageFlow should have a messageRef property defined.
Total Errors : 4
Total Warnings : 0
Total Fixes : 2
VALIDATING DIAGRAM <Activity4> OF <Activity4>
Error on Activity <Activity6>: [BIR_30] => An Activity, Start Event, End Event, or Intermediate Event of type Message should have a messageRef property defined.
Error on Activity <Activity5>: [BIR_13] => A send Task should have an outgoing Message Flow.
Error on Activity <Activity5>: [BIR_30] => An Activity, Start Event, End Event, or Intermediate Event of type Message should have a messageRef property defined.
Error on connector <MessageFlow>: [BIR_08] => The Message Flow should be labeled with the name of the message it is carrying.
Error on connector <MessageFlow>: [BIR_38] => The MessageFlow should have a messageRef property defined.
Total Errors : 5
Total Warnings : 0
Total Fixes : 0
  
```

Each active line item in the log contains:

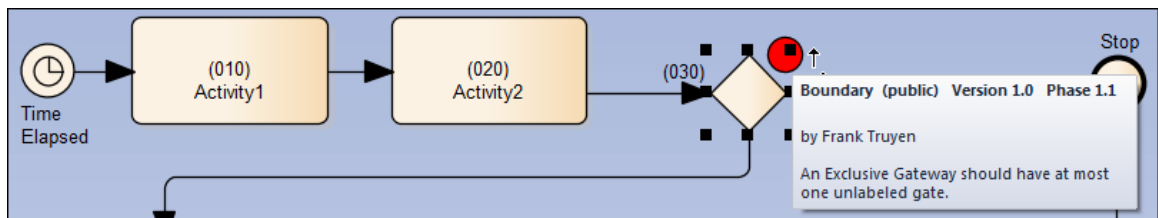
- An Error or warning message including the rule ID (see [documentation below](#)).
- The relevant element (e.g. Activity) or connector (e.g. Sequence Flow) type.

If an automatic fix was applied, an inactive child line item is added to document the change (see examples in the above screenshot).

A single mouse click on an active error or warning line in the Output window will automatically:

- Select the related element (if any) in the Project Browser.
- Make the diagram in which the element resides the currently active context (i.e. make it visible).
- Select the element (if any) in the diagram. In EA 13 the selected diagram element will be automatically centered for easy viewing. In EA 12.1, the diagram window may need to be scrolled to view the element.
- Select the connector (if any) in the diagram. The diagram window may need to be scrolled to view the connector.
- For errors/warnings relating to elements, a small red marker is added near the top right corner of the element to draw attention to it, and to retain a copy of the error/warning message after the Output tab content has been cleared. Please note that adding markers is currently not possible for connectors. The markers are not displayed if:
 - The Package containing the process model is locked.
 - The current license is a trial version.
 - The [option has been disabled](#).

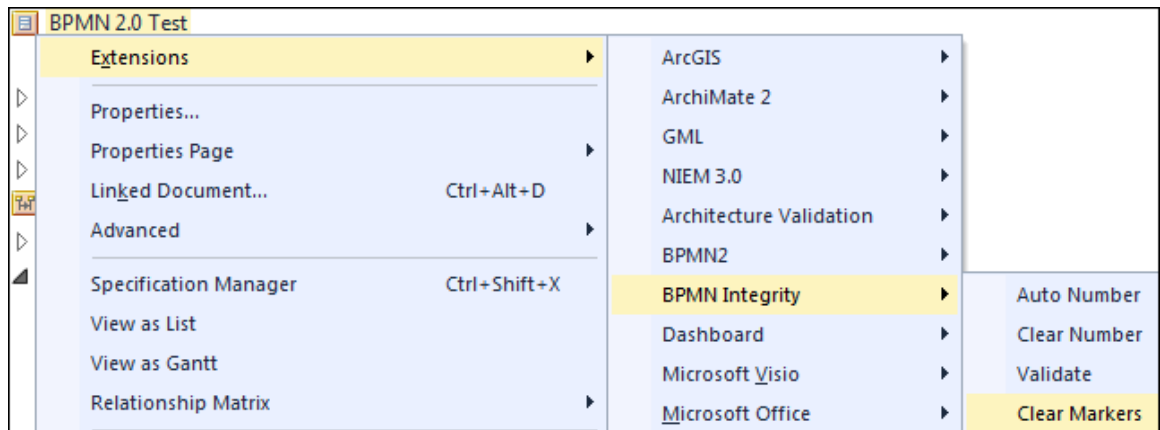
Diagram marker example



Moving the mouse over the marker will display the related error/warning message.

Markers are deleted automatically (if the parent Package is unlocked) by running subsequent validations against the same model context.

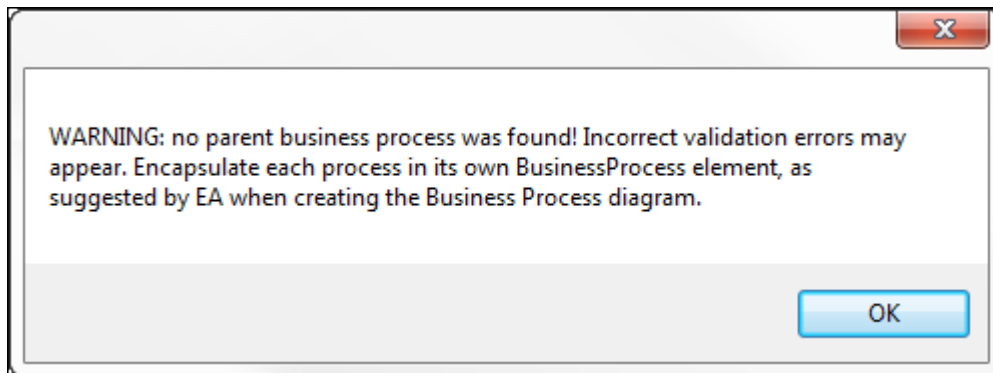
They can also be removed from the context of the parent Package by selecting:



A double click action on an active line item in the log opens the properties of the related element (if any). This is currently not possible for connectors.

Note that for error/warning messages that do not relate to a specific element or connector in the repository the above actions will not take place (i.e. the line item in the log will be inactive).

When starting the validation, the following message may appear:



The validation will proceed, but some erroneous rule infractions may be reported.

To avoid this situation, and to be in better compliance with the BPMN 2.0 specification, always create a Business Process element as the container of your process flow.

If Pools are used:

- a) Allow the tool to automatically create a Collaboration Model element when adding the first Pool to the diagram.
- b) If the internal process flow elements are modeled inside a Pool (with or without additional Lanes), set the *processRef* tagged value of that Pool to reference the Business Process element. Doing so makes it clear which diagram elements are internal to the process.
- c) For Pools representing external participants, either leave them empty (as "black-boxes"), or if flow elements are represented inside them, set the

BPMN Integrity User Guide

processRef tagged value of each Pool to the corresponding Business Process where these elements reside (every Pool relates to a single Business Process).

Default Rule Set

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE Validations >
<Rules Version="1" AddMarkers="True" MaxActivityCount="20">
  <BIR_01 Enabled="True" AutoFix="True" />
  <BIR_02 Enabled="True" />
  <BIR_03 Enabled="True" AutoFix="True" />
  <BIR_04 Enabled="True" />
  <BIR_05 Enabled="True" AutoFix="True" />
  <BIR_06 Enabled="True" />
  <BIR_07 Enabled="True" />
  <BIR_08 Enabled="True" AutoFix="True" />
  <BIR_09 Enabled="True" />
  <BIR_10 Enabled="True" />
  <BIR_11 Enabled="True" />
  <BIR_12 Enabled="True" />
  <BIR_13 Enabled="True" />
  <BIR_14 Enabled="True" />
  <BIR_15 Enabled="True" AutoFix="True" />
  <BIR_16 Enabled="True" />
  <BIR_17 Enabled="True" />
  <BIR_18 Enabled="True" />
  <BIR_19 Enabled="True" />
  <BIR_20 Enabled="True" />
  <BIR_21 Enabled="True" />
  <BIR_22 Enabled="True" />
  <BIR_23 Enabled="True" AutoFix="True" />
  <BIR_24 Enabled="True" />
  <BIR_25 Enabled="True" />
  <BIR_26 Enabled="True" />
  <BIR_27 Enabled="True" />
  <BIR_28 Enabled="True" />
  <BIR_29 Enabled="True" AutoFix="True" />
  <BIR_30 Enabled="True" AutoFix="True" />
  <BIR_31 Enabled="True" />
  <BIR_32 Enabled="True" />
  <BIR_33 Enabled="True" />
  <BIR_34 Enabled="True" />
  <BIR_35 Enabled="True" />
  <BIR_36 Enabled="True" AutoFix="True" />
  <BIR_37 Enabled="True" AutoFix="True" />
  <BIR_38 Enabled="True" AutoFix="True" />
  <BIR_39 Enabled="True" AutoFix="True" />
  <BIR_40 Enabled="True" AutoFix="True" />
  <BIR_41 Enabled="True" />
  <BIR_42 Enabled="True" />
  <BIR_43 Enabled="True" />
  <BIR_44 Enabled="True" />
  <BIR_45 Enabled="True" />
  <BIR_46 Enabled="True" />
  <BIR_47 Enabled="True" />
</Rules>
```

In the default rule set:

- All rules are enabled.
- All auto-fix options are enabled.
- The option to [add diagram markers](#) is enabled.
- The maximum number of Activities per diagram is set to 20 (see rule [BIR_35](#)).

Customizing the Rule Set

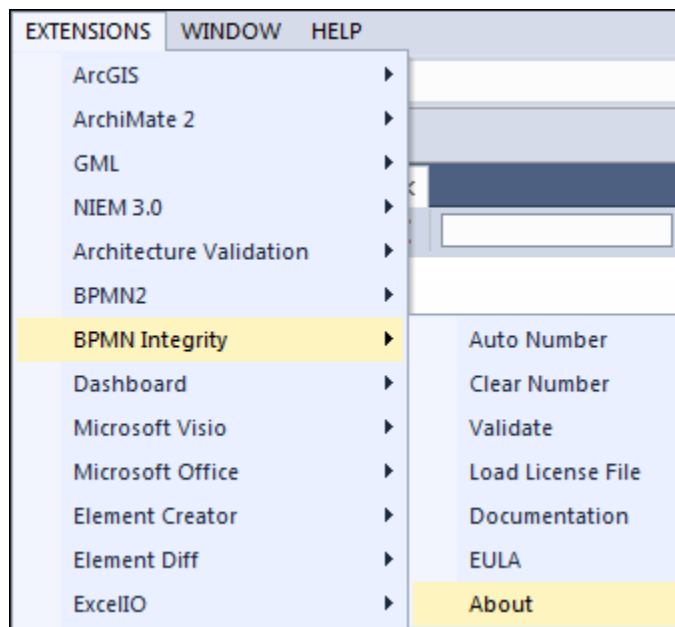
Note: this option is not available in the trial version!

Caveat: it is strongly recommended to save a copy of the default rule set before making any modifications!

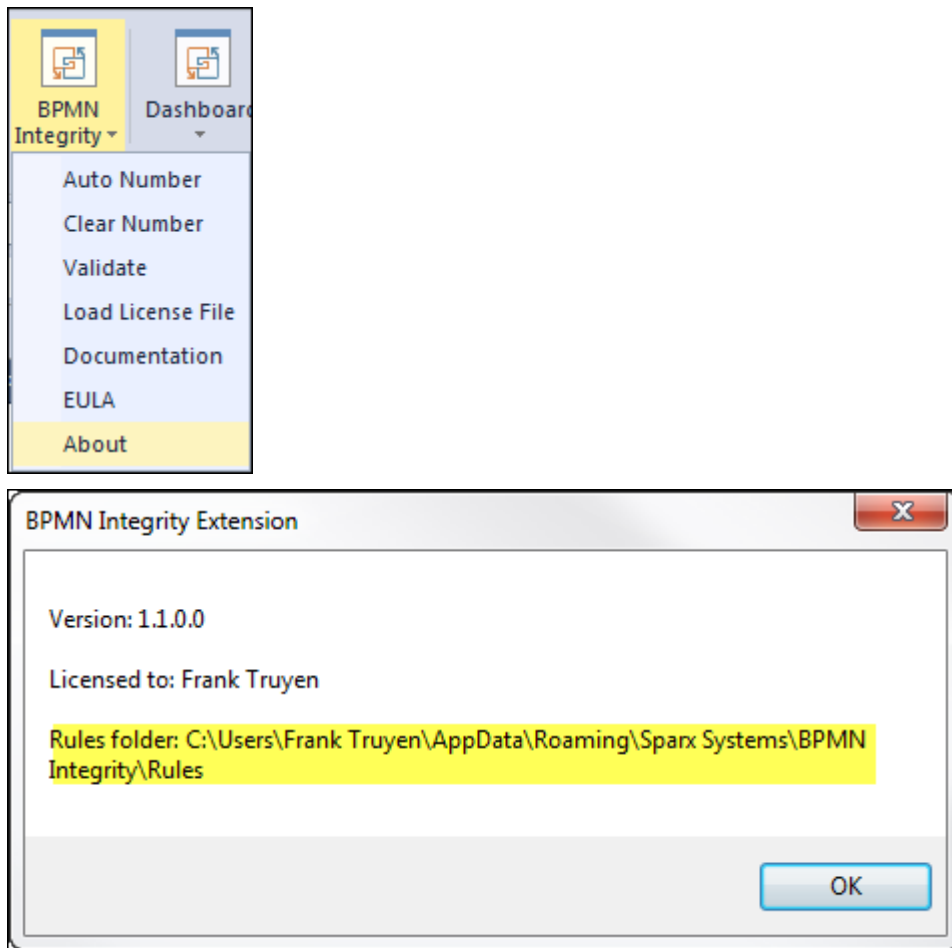
Open the *IntegrityRules.xml* file in any XML editor and make the required changes.

To determine the location of the rules folder on your system, select the About menu item.

In EA 12.1:



In EA 13.x, in the EXTEND ribbon:



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

Integrity Rules

Rule BIR_01

Description	A Start Event of type other than None should be labeled to indicate the process trigger.
Motivation	Makes it easier to understand what triggers the process flow.
Child messages	None
Auto-fix	<p>Yes in these cases:</p> <ul style="list-style-type: none"> • A Conditional Start Event with a <i>conditionExpression</i> property defined: the name is set to the same value as the expression. • A Timer Start Event with a time cycle or time duration property value defined: the name is set to the value. • An Error Start Event with an <i>errorRef</i> property defined: the name is set to the same value as the referenced Error element. • An Escalation Start Event with an <i>escalationRef</i> property defined: the name is set to the same value as the referenced Escalation element. • A Signal Start Event with a <i>signalRef</i> property defined: the name is set to the same value as the referenced Signal element.

Rule BIR_02

Description	For compliance with the BPMN 2.0 XML specification, a Pool should be the child of a <i>CollaborationModel</i> element.
Motivation	Compliance with the BPMN 2.0 XML specification is important in case the process needs to be exported to another tool, for example a simulation or process execution engine.
Child messages	None
Auto-fix	<p>No.</p> <p>Manually create a <i>CollaborationModel</i> element (from the Collaboration toolbox) at the same level as the <i>BusinessProcess</i> element. Leave the diagram unchanged, but in the Project Browser move the Pool (and all of its content) under the <i>CollaborationModel</i> element.</p>

Rule BIR_03

Description	A Lane should be named, or have its <i>partitionElementRef</i> property set to a defined role.
Motivation	Named Lanes identify who is responsible for performing the activities modeled inside them.
Child messages	None
Auto-fix	No. If the <i>partitionElementRef</i> property is set to a defined role element (e.g. a BPMN ResourceRole) then the tool will automatically display that role name as the Lane's label in the diagram (but not in the Project Browser), unless overwritten by the User in the name property.

Rule BIR_04

Description	A Message Start Event should be labeled "Receive <message name>".
Motivation	Clarifies which message is expected to trigger the process.
Child messages	None
Auto-fix	No.

Rule BIR_05

Description	An Intermediate Event should be labeled to indicate the event trigger.
Motivation	Makes it easier to understand what triggers the event.
Child messages	None
Auto-fix	<p>Yes in these cases:</p> <ul style="list-style-type: none"> • A Conditional Intermediate Event with a <i>conditionExpression</i> property defined: the name is set to the same value as the expression. • A Timer Intermediate Event with a time cycle or time duration property value defined: the name is set to the value. • An Error Intermediate Event with an <i>errorRef</i> property defined: the name is set to the same value as the referenced Error element. • An Escalation Intermediate Event with an <i>escalationRef</i> property defined: the name is set to the same value as the referenced Escalation element. • A Signal Intermediate Event with a <i>signalRef</i> property defined: the name is set to the same value as the referenced Signal element. • A Message Intermediate Event with a <i>messageRef</i> property defined: the name is set to the same value as the referenced Message, prefixed with "Receive".

Rule BIR_06

Description	An Intermediate Event of type Link must reference a matching Link Event in the same scope.
Motivation	For a throw or catch Link Event to be valid, a corresponding catch or throw Link Event must exist within the same process scope. Link Events are not allowed to cross Process or Sub-Process boundaries.
Child messages	
Description	The corresponding Link Intermediate Event matches this Link Event on the name, but not on the <i>linkEventSources/linkEventTarget</i> property value.
Description	The corresponding Link Intermediate Event matches this Link Event <i>linkEventSources/linkEventTarget</i> property values, but not on the reverse property values.
Auto-fix	No.

Rule BIR_07

Description	An Activity should be labeled.
Motivation	
Child messages	None
Auto-fix	No.

Rule BIR_08

Description	The Message Flow should be labeled with the name of the message it is carrying.
Motivation	Clarifies what data is being conveyed.
Child messages	
Description	The Message Flow label [message 1] does not match the <i>messageRef</i> property [message 2] specified in [sending or receiving element name, or the Message Flow itself].
Description	There are conflicting <i>messageRef</i> properties defined in the Message Flow and/or the source element and/or the target element!
Auto-fix	Yes. If a <i>messageRef</i> property is defined in the Message Flow, the sending element, or receiving element, and these Message references do not conflict with one another, then the label is set to the name of the referenced Message.

Rule BIR_09

Description	A Message should be labeled.
Motivation	A Message element should be labeled so that its name can be resolved when referenced through a <i>messageRef</i> property.
Child messages	None
Auto-fix	No.

Rule BIR_10

Description	A Message Start Event should have an incoming Message Flow.
Motivation	The Start Event may have a <i>messageRef</i> property defined to document the Message expected, but a Message Flow is required to indicate from whom the Message will be received.
Child messages	None
Auto-fix	No.

Rule BIR_11

Description	A catching Message Intermediate Event should have an incoming Message Flow.
Motivation	The Intermediate Event may have a <i>messageRef</i> property defined to document the Message expected, but a Message Flow is required to indicate from whom the Message will be received.
Child messages	None
Auto-fix	No.

Rule BIR_12

Description	A throwing Message Intermediate or End Event should have an outgoing Message Flow.
Motivation	The Event may have a <i>messageRef</i> property defined to document the Message to be sent, but a Message Flow is required to indicate to whom the Message will be delivered.
Child messages	None
Auto-fix	No.

Rule BIR_13

Description	A Send Task should have an outgoing Message Flow.
Motivation	The Task may have a <i>messageRef</i> property defined to document the Message to be sent, but a Message Flow is required to indicate to whom the Message will be delivered.
Child messages	None
Auto-fix	No.

Rule BIR_14

Description	A Receive Task should have an incoming Message Flow.
Motivation	The Task may have a <i>messageRef</i> property defined to document the Message expected, but a Message Flow is required to indicate from whom the Message will be received.
Child messages	None
Auto-fix	No.

Rule BIR_15

Description	An Exclusive or Inclusive Gateway should have at most one outgoing default Sequence Flow.
Motivation	Only one outgoing Sequence Flow can be marked as the default path in case all other Sequence Flow conditions evaluate to false.
Child messages	
Description	The outgoing Gateway Sequence Flow has an unlabeled gate which is not set to default.
Description	Unable to update the <i>conditionType</i> property of the unlabeled Sequence Flow to Default. There is already a default Flow defined!
Auto-fix	Yes. If an unlabeled outgoing Sequence Flow is detected, and no other outgoing Sequence is marked as default, then the <i>conditionType</i> property is automatically set to default.

Rule BIR_16

Description	An Exclusive or Inclusive Gateway with an unlabeled outgoing Sequence Flow should itself be labeled.
Motivation	This helps the reviewer/reader to understand under which conditions the outgoing default Sequence Flow path may be taken.
Child messages	None
Auto-fix	No.

Rule BIR_17

Description	A non-ad-hoc Activity (other than an Event Sub-process) should have at least one incoming and one outgoing Sequence Flow.
Motivation	Good process modeling practice prescribes the use of a Start Event to indicate the beginning of the flow, and of at least one End Event to mark the end of the flow.
Child messages	
Description	Conditional outgoing Sequence Flows of an Activity should be replaced by an Exclusive Gateway.
Motivation	Gateways are graphically more explicit and easier to understand by the consumer of the model.
Description	Multiple unconditional outgoing Sequence Flows of an Activity should be replaced by a Parallel Gateway to better document the concurrent processing.
Motivation	Gateways are graphically more explicit and easier to understand by the consumer of the model.
Auto-fix	No.

Rule BIR_18

Description	Two End Events that are part of the same scope should not have the same name. If they mean the same end state, combine them; otherwise give them different names.
Motivation	Clarifies the conditions under which a process flow ends.
Child messages	None
Auto-fix	No.

Rule BIR_19

Description	If there is more than one End Event in the same scope, each one should be labeled with the name of an end state.
Motivation	If there is more than one End Event present in the same scope, name each one to clarify the conditions under which a process flow ends.
Child messages	None
Auto-fix	No.

Rule BIR_20

Description	An Error Intermediate Event on the boundary of a Sub-Process should correspond to an Error End Event in the process, either by its name or by its <i>errorRef</i> property.
Motivation	Removes any ambiguity as to which Error is being caught by the event handler. BPMN 2.0 does allow the use of “catch all” event handlers, however this is not a recommended practice since it can lead to incorrect handling of error conditions as the model changes over time.
Child messages	
Description	The Error End Event, corresponding to a boundary Intermediate Error Event, matches it on the name, but not on the <i>errorRef</i> property value.
Description	The Error End Event, corresponding to a boundary Intermediate Error Event, matches it on the <i>errorRef</i> property value, but not on the name.
Description	An Intermediate Event of type Error must be on the boundary of a Sub-Process that throws a corresponding Error End Event.
Motivation	An Intermediate Event of type Error should not be on the boundary of a Task since the event would never be triggered.
Auto-fix	No.

Rule BIR_21

Description	An Escalation Intermediate Event on the boundary of a Sub-Process should correspond to an Escalation End or Intermediate Event in the process, either by its name or by its <i>esacalationRef</i> property.
Motivation	Removes any ambiguity as to which Escalation is being caught by the event handler. BPMN 2.0 does allow the use of “catch all” event handlers, however this is not a recommended practice since it can lead to incorrect handling of escalation conditions as the model changes over time.
Child messages	
Description	The Escalation Event, corresponding to a boundary Intermediate Event, matches it on the name, but not on the <i>esacalationRef</i> property value.
Description	The Escalation Event, corresponding to a boundary Intermediate Event, matches it on the <i>esacalationRef</i> property value, but not on the name.
Description	An Intermediate Event of type Escalation must be on the boundary of a Sub-Process that throws a corresponding Escalation Intermediate or End Event.
Motivation	An Intermediate Event of type Escalation should not be on the boundary of a Task since the event would never be triggered.
Auto-fix	No.

Rule BIR_22

Description	Two Activities in the same scope should not have the same name.
Motivation	
Child messages	None
Auto-fix	No.

Rule BIR_23

Description	The label of a child-level diagram should match the name of the parent Sub-Process.
Motivation	Consistency.
Child messages	None
Auto-fix	Yes. The diagram name is updated to match the Sub-Process name.

Rule BIR_24

Description	Only one Start Event of type None should be used in a Sub-Process.
Motivation	A Sub-Process may contain additional Start Events, to make the Sub-Process independently callable, but these Start Events must have a trigger other than None.
Child messages	None
Auto-fix	No.

Rule BIR_25

Description	If a Sub-Process is followed by an Exclusive Gateway, the label of at least one End Event of the Sub-Process should match the label/condition of each outgoing Sequence Flow of the Gateway.
Motivation	Correlating the Sub-Process End Events with the Gateway's outgoing Sequence Flow conditions provides an easy way to ensure all Sub-Process exit points are properly handled after the process completes. See also rule BIR_27 .
Child messages	
Description	No End Event found in the sub-process preceding the Exclusive Gateway matches the label of this Sequence Flow.
Auto-fix	No.

Rule BIR_26

Description	Child level elements should not be enclosed in an expanded Sub-Process. Use a composite child diagram instead.
Motivation	While expanded Sub-Processes are allowed in BPMN 2.0, good modeling practice recommends to use a child (composite) diagram instead. It avoids having to make complex diagram adjustments when the Sub-Process steps grow in number, and helps reduce the size of the parent diagram.
Auto-fix	No.

Rule BIR_27

Description	If a Sub-Process is followed by an exclusive Gateway it should have an End Event for each non-default Sequence Flow following the Gateway.
Motivation	This is the corollary to rule BIR_25 : the number of non-default Sequence Flows should match the number of End Events in the Sub-Process.
Auto-fix	No.

Rule BIR_28

Description	A Start, End, or Intermediate Event of type Signal should have the <i>signalRef</i> property defined.
Motivation (warning only)	Explicitly associating the Signal Event with a reference to a Signal element provides for better traceability across the model.
Child messages	
Description	The referenced Signal [signal name] is not thrown anywhere in this repository!
Motivation (warning only)	Alerts the modeler that no correlating throw event with the same Signal name (or <i>signalRef</i> property) can currently be found in the database. If the source of the event is an external business process, or is not modeled in the current repository, then this should not be considered an error.
Auto-fix	No.

Rule BIR_29

Description	A Start or Intermediate Event of type Conditional should have its <i>condition</i> property defined.
Motivation	The <i>condition</i> property allows the rule to be specified in a more formal language (or precise manner) than just the name of the Event.
Auto-fix	Yes. If the name of the Event is not blank it is copied into the <i>condition</i> property.

Rule BIR_30

Description	An Activity, Start Event, End Event, or Intermediate Event of type Message should have its <i>messageRef</i> property defined.
Motivation	The <i>messageRef</i> property allows the unambiguous identification of the Message being sent or received.
Auto-fix	Yes, under these conditions: <ul style="list-style-type: none"> • A <i>messageRef</i> property is specified either in the incoming or outgoing Message Flow. • Or a <i>messageRef</i> property is specified in the element on the other side of the Message Flow. • The above <i>messageRef</i> properties do not conflict.

Rule BIR_31

Description	No outgoing Message Flows were found in the child diagram that replicate the outgoing Message Flows of the collapsed Sub-Process.
Motivation	An outgoing Message Flow of a collapsed Sub-Process cannot originate from the Sub-Process element itself. It is thus important to identify inside the Sub-Process which Send Task, Message End Event, or Message Intermediate Event (of type throw) is responsible for generating the Message. See also rule BIR_32 .
Child messages	
Description	No match on name or <i>messageRef</i> property was found in the outgoing Message Flows of the child diagram that replicate the outgoing Message Flows of the collapsed Sub-Process.
Auto-fix	No.

Rule BIR_32

Description	No incoming Message Flows were found in the child diagram that replicate the incoming Message Flows from the collapsed Sub-Process.
Motivation	An incoming Message Flow of a collapsed Sub-Process cannot be received by the Sub-Process element itself. It is thus important to identify inside the Sub-Process which Receive Task, Message Start Event, or Message Intermediate Event (of type catch) is responsible for receiving the Message. See also rule BIR_31 .
Child messages	
Description	No match on name or <i>messageRef</i> property was found in the incoming Message Flows of the child diagram that replicate the incoming Message Flows of the collapsed Sub-Process.
Auto-fix	No.

Rule BIR_33

Description	One or more incoming Message Flows were detected in the child diagram that are not replicated in the parent level diagram of the collapsed Sub-Process.
Motivation	For consistency between the high level (collapsed) view of the Sub-Process and its detailed view (in the child diagram), Message Flows should be represented in both diagrams. See also rule BIR_34 .
Auto-fix	No.

Rule BIR_34

Description	One or more outgoing Message Flows were detected in the child diagram that are not replicated in the parent level diagram of the collapsed Sub-Process.
Motivation	For consistency between the high level (collapsed) view of the Sub-Process and its detailed view (in the child diagram), Message Flows should be represented in both diagrams. See also rule BIR_33 .
Auto-fix	No.

Rule BIR_35

Description	This diagram contains more than <i>n</i> process steps. Consider using (additional) sub-processes.
Motivation	To make process flows easier to review and understand, it is highly recommended not to exceed a certain number of Activities per process scope. The maximum defined in the default rule set is 20, but this number can be adjusted by the User .
Auto-fix	No.

Rule BIR_36

Description	An Activity modeled as Sub-Process should have its <i>activityType</i> property set to 'Sub-Process'.
Motivation	If an Activity has been modeled as either a collapsed or expanded Sub-Process, then its <i>activityType</i> should be set accordingly.
Auto-fix	Yes.

Rule BIR_37

Description	A Sub-Process should have its <i>taskType</i> property set to 'Abstract'.
Motivation	<i>taskType</i> values other than 'Abstract' apply to Tasks only, not to Sub-Processes.
Auto-fix	Yes.

Rule BIR_38

Description	A Message Flow should have its <i>messageRef</i> property defined.
Motivation	The <i>messageRef</i> property allows the unambiguous identification of Message being sent or received.
Auto-fix	Yes, under these conditions: <ul style="list-style-type: none"> • A <i>messageRef</i> property is specified in the source and/or the target element of the Message Flow. • The above <i>messageRef</i> properties do not conflict.

Rule BIR_39

Description	A non-default Sequence Flow out of an Inclusive or Exclusive Gateway should have its <i>conditionExpression</i> property defined.
Motivation	Setting this property is the required way in BPMN 2.0 to express the condition to be evaluated before a conditional Sequence Flow path is taken. Its value is automatically displayed as the Sequence Flow label.
Auto-fix	Yes. If the name/label field of Sequence Flow has been defined then its value is copied to the <i>conditionExpression</i> property. In addition the <i>conditionType</i> property is set to "Expression" if it does not have this value already. Finally the name field of the Sequence Flow is changed to blank, to avoid having duplicate labels shown in the diagram.

Rule BIR_40

Description	If a <i>conditionExpression</i> property is specified in an outgoing Sequence Flow of an Inclusive or Exclusive Gateway, then its <i>conditionType</i> property must be set to 'Expression'.
Motivation	For BPMN 2.0 compliance. This is verified by the BPMN 2.0 syntax checker as well, but this rule will auto-fix the issue.
Auto-fix	Yes.

Rule BIR_41

Description	An Error Start Event in an Event Sub-Process should correspond to an Error End Event in the process, either by its name or by its <i>errorRef</i> property.
Motivation	Removes any ambiguity as to which Error is being caught by the event handler. BPMN 2.0 does allow the use of “catch all” event handlers, however this is not a recommended practice since it can lead to incorrect handling of error conditions as the model changes over time.
Child messages	
Description	The Error End Event, corresponding to an Event Sub-Process Start Error Event, matches it on the name, but not on the <i>errorRef</i> property value.
Description	The Error End Event, corresponding to an Event Sub-Process Start Error Event, matches it on the <i>errorRef</i> property value, but not on the name.
Auto-fix	No.

Rule BIR_42

Description	No matching event handler was found within the validation scope for this [event kind] of type [event type].
Motivation	<p>This rule applies to:</p> <ul style="list-style-type: none"> • End Events of type Error, Cancel, Escalation. • Intermediate throw Events of type Escalation. <p>For any of these Events, a corresponding catch handler should be defined somewhere in the preceding (higher level) process hierarchy, otherwise the Event is unhandled.</p>
Auto-fix	No.

Rule BIR_43

Description	A Start, Intermediate or End Event of type Error should have its <i>errorRef</i> property defined.
Motivation	The <i>errorRef</i> property allows the unambiguous identification of the Error being thrown or caught.
Auto-fix	No.

Rule BIR_44

Description	A Start, Intermediate or End Event of type Escalation should have its <i>escalationRef</i> property defined.
Motivation	The <i>escalationRef</i> property allows the unambiguous identification of the Escalation being thrown or caught.
Auto-fix	No.

Rule BIR_45

Description	No Cancel End Event was found in the Sub-Process which can trigger this boundary Cancel Intermediate Event.
Motivation	If a Cancel Intermediate Event handler is put on the boundary of a Sub-Process that process must have a corresponding Cancel End Event to trigger the handler.
Child messages	
Description	No Cancel End Event was found in the Sub-Process with a name matching this boundary Cancel Intermediate Event.
Auto-fix	No.

Rule BIR_46

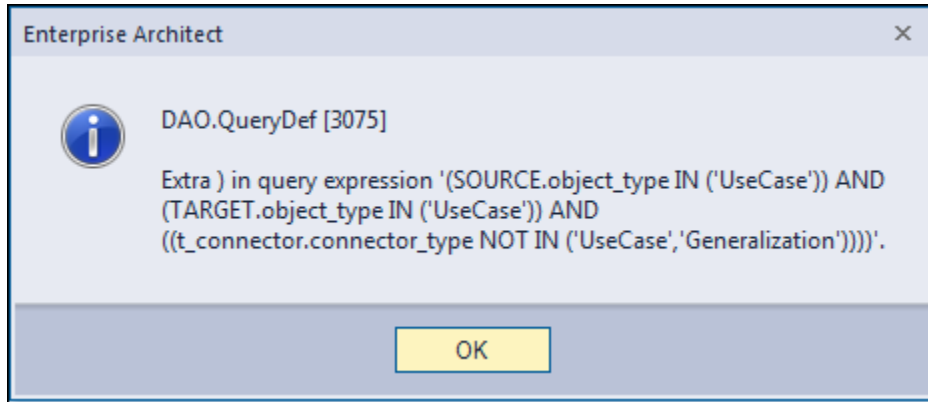
Description	An Escalation Start Event in an Event Sub-Process should correspond to an Escalation End or Intermediate Event in the process, either by its name or by its <i>esacalationRef</i> property.
Motivation	Removes any ambiguity as to which Escalation is being caught by the Event Sub-Process. BPMN 2.0 does allow the use of “catch all” event handlers, however this is not a recommended practice since it can lead to incorrect handling of escalation events as the model changes over time.
Child messages	
Description	The Escalation Event, corresponding to an Event Sub-Process Start Escalation Event, matches it on the name, but not on the <i>esacalationRef</i> property value.
Description	The Escalation Event, corresponding to an Event Sub-process Start Escalation Event, matches it on the <i>esacalationRef</i> property value, but not on the name.
Auto-fix	No.

Rule BIR_47

Description	No Cancel End Event was found in the process which can trigger this Event Sub-Process Cancel Start Event.
Motivation	If an Event Sub-Process is triggered by a Cancel Start Event, its parent process must have a corresponding Cancel End Event somewhere in its scope to trigger the handler.
Child messages	
Description	No Cancel End Event was found in the process with a name matching this Event Sub-Process Cancel Start Event.
Auto-fix	No.

Troubleshooting

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



As of versions 12.1 and 13.x of Enterprise Architect this type of error is not relayed back to the extension which is unaware that a problem occurred.

Please follow this procedure:

- Take a screenshot of the error message.
- Export a copy of the model being validated to XML.
- Locate the DBError.txt file in %APPDATA%\Sparx Systems\EA and include it in your message.
- Provide your database type (Microsoft Access, SQL Server, Oracle, etc.) and version number.
- Include the Enterprise Architect version.
- Email the above information to the support address listed below.

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 : 714-573-7112.