Activity diagram elements

On this page

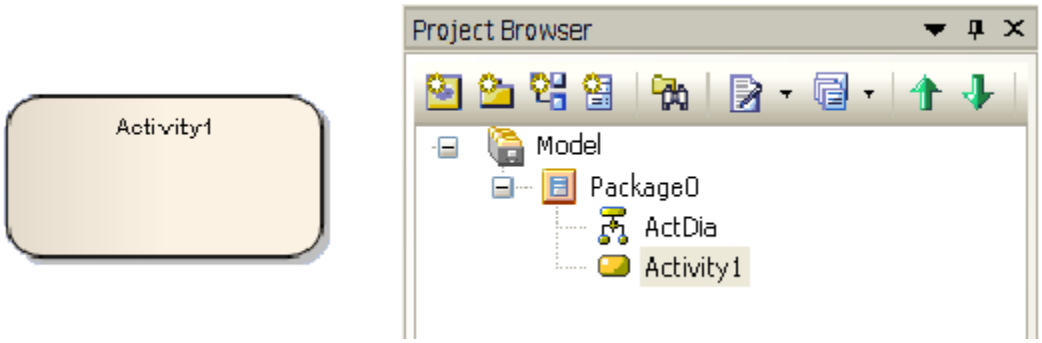
Activity

You can directly place an Activity element in EA as an element view on an Activity diagram. However, this behavior conflicts with MagicDraw (CEA and CSM) and UML notation. In MagicDraw, CEA, or CSM, if you drag an Activity from the containment tree to an Activity diagram, a new CallBehaviorAction vi ew will be created and the **Behavior** property of the CallBehaviorAction will be set to the Activity. This same behavior will be used in the import process.

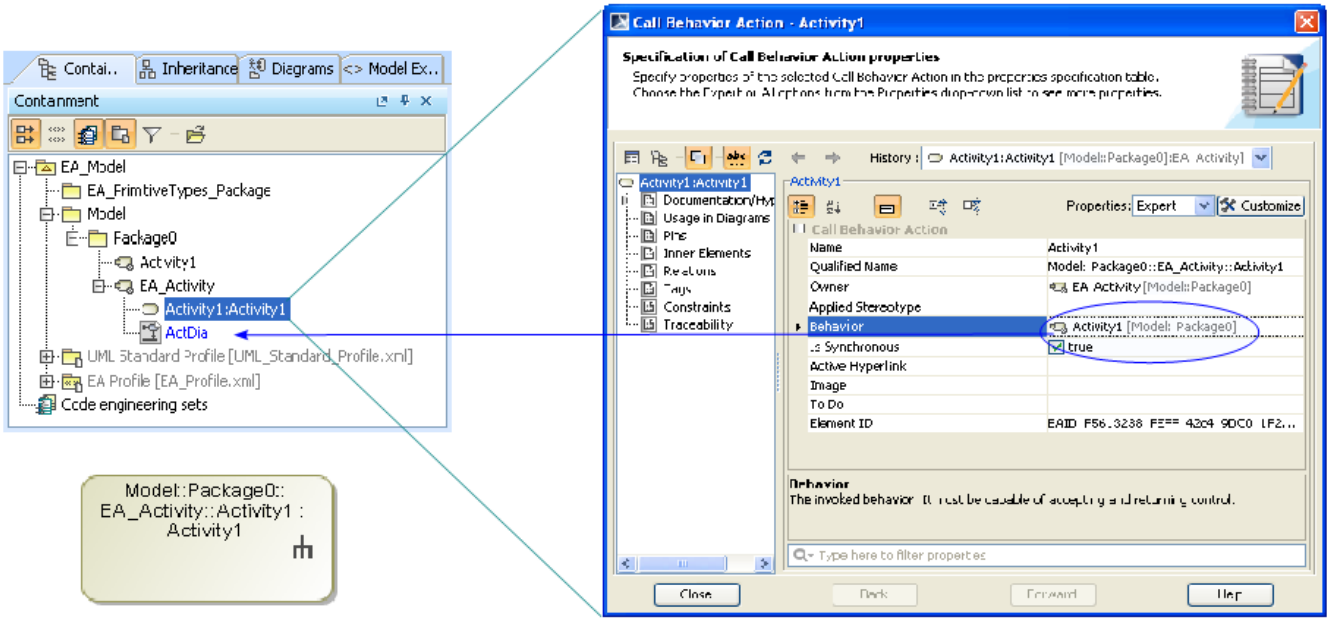
An Activity element created in EA and placed on an Activity diagram will be transformed into two elements: Activity, and CallBehaviorAction elements. Both elements will have the same name and will be linked through the property of a CallBehaviorAction element called *Behavior*.

After transforming the element, the following transformation message opens:

> *Updated element <xmi:id>: A new CallBehaviorAction was created and its Behavior was set to the element.*



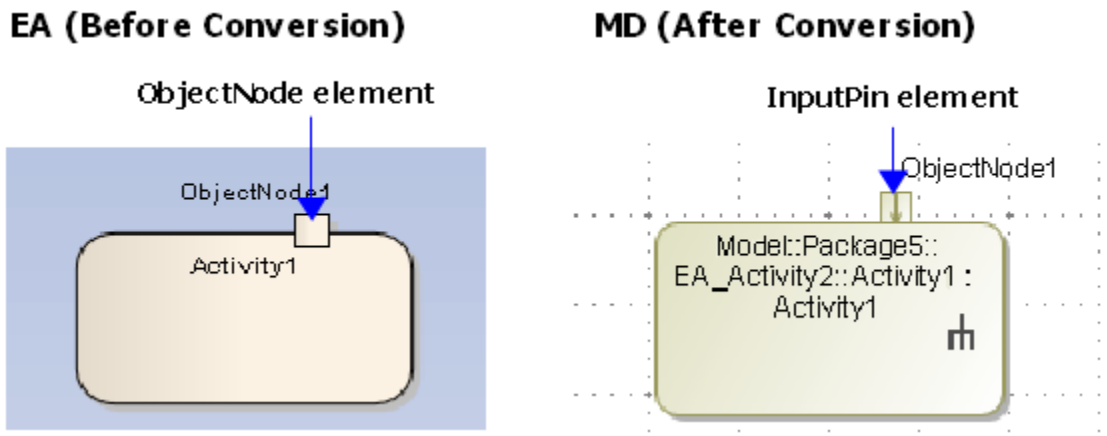
EA activity.



Activity with a New CallBehavior in MagicDraw.

> **Note**
> The EA Activity and CallBehaviorAction elements have similar characteristics in that you can attach a control flow to it and others. EA has its own CallBehaviorAction element.

Additionally, any ObjectNode elements attached to the Activity element will be transformed into InputPin elements and attached to the newly created CallBehaviorAction element.



An ObjectNode converted into an InputPin.

Activity diagram

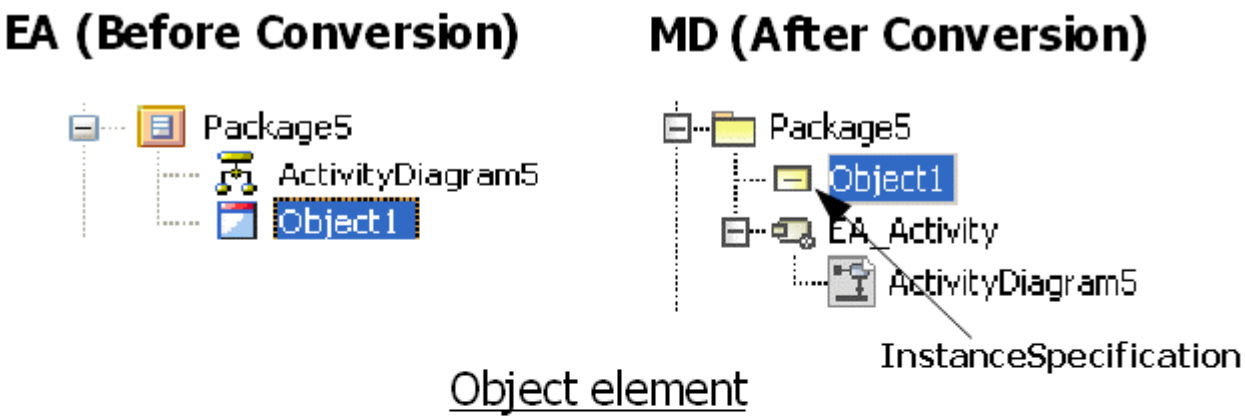
Every Activity Diagram element from EA will be placed inside an Activity element that has the same name.

Activity diagram in the MagicDraw Containment tree.

Note
In MagicDraw, CEA, or CSM, every Activity diagram element must be placed inside an Activity element that has the same name. However, this is not the case in EA.

Object as the inner element of an Activity

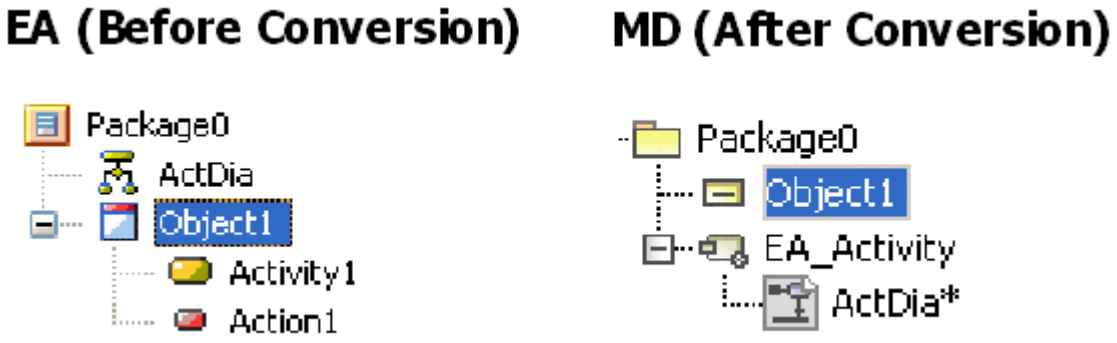
Object elements inside an Activity element in EA will be removed.



Object elements transformation.

Note
Object elements in MagicDraw, CEA, or CSM have their XMI types defined as *uml:Central-BufferNode*. However, those in EA have their XMI types defined as *uml:InstanceSpecification*, which do not belong to an Activity diagram.

An Object element containing any ActivityDiagram-related elements will be removed.

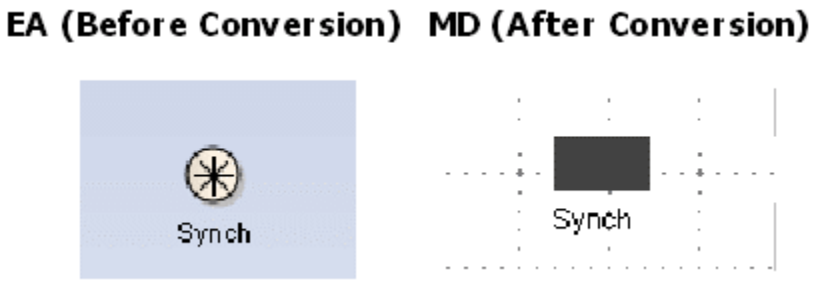


Object Containing Activity-related Elements.

Note
In MagicDraw, CEA, or CSM, an Object element (CentralBufferNode) is not allowed to contain elements other than comments and hyperlinks.

Synch Node

A Synch element in EA will be transformed into a Join element in MagicDraw, CEA, or CSM. It will look exactly like a Fork/Join element.



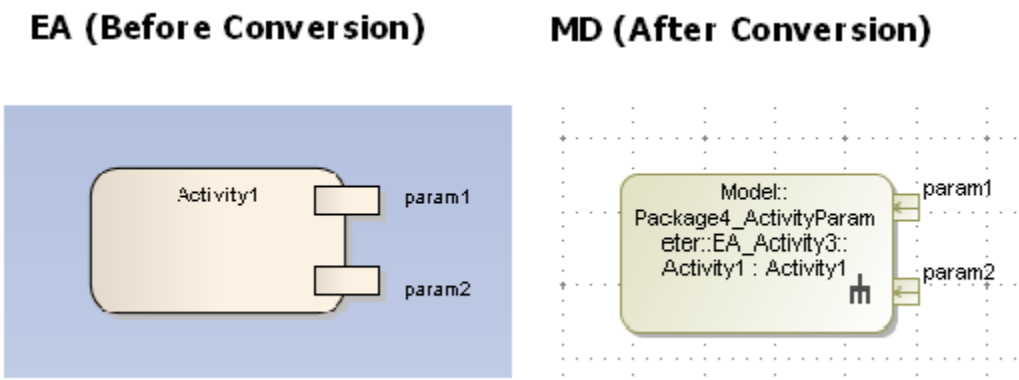
Synch element transformation.

Note

A MagicDraw (CAE or CSM) Fork/Join element (whose type is *uml:ForkNode*) can be used to construct either a Fork and Join node in an Activity diagram. The JoinNode element (whose type is *uml:JoinNode*) is allowed to be placed in the Activity diagram, but the element's image will be displayed as the Fork/Join element's default image.

Activity Parameter

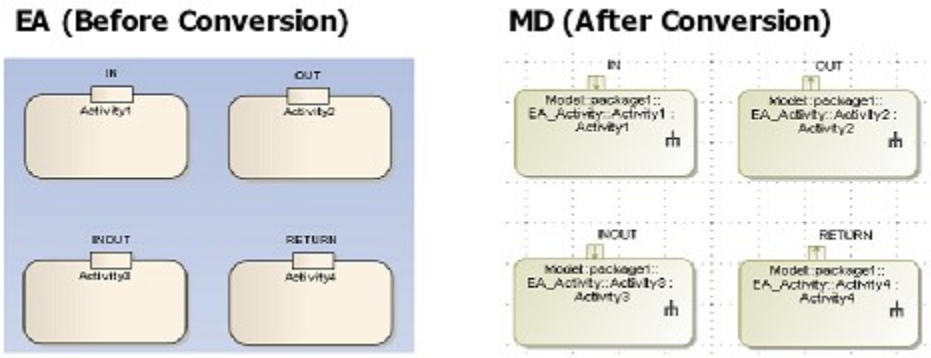
If you create an ActivityParameter element, MagicDraw (CEA or CSM) will automatically create an ActivityParameterNode element to represent it. Every ActivityParameterNode element in EA will be transformed into a Pin element.



Activity parameter node.

You can specify four parameter types for each Activity Parameter element: in, out, inout, and return.

The ActivityParameterNode element of an ActivityParameter element whose parameter type is either 'in' or 'inout' will be transformed into an InputPin element. The ActivityParameterNode element of an ActivityParameter element whose parameter type is either *out* or *return* will be transformed into an OutputPin element.



Activity parameter type.

Note

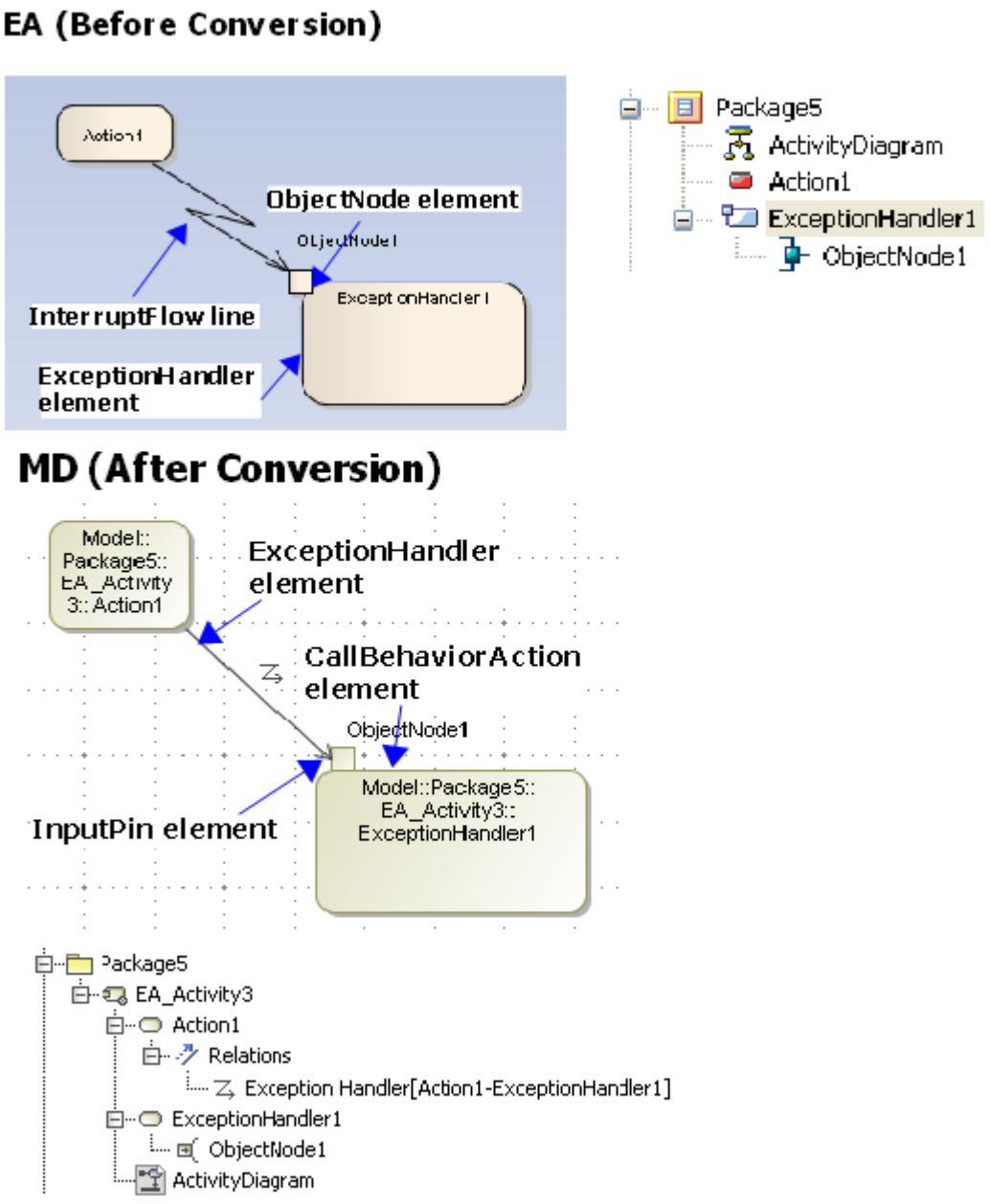
Usually, if you specify the parameter type of an ActivityParameter element as *inout*, two Pin elements (InputPin and OutputPin elements) will be created for the element. Since EA will only create one ActivityParameterNode element, this element will be transformed into an InputPin element.

Exception Handler

The Exception Handler element in EA differs from the UML's ExceptionHandler. This EA element will be transformed into a CallBehaviorAction element. Any ObjectNode element attached to it will be transformed into an InputPin element and any InterruptFlow line will be transformed into an ExceptionHandler line in MagicDraw, CEA, or CSM.

After completing the transformation, the following transformation messages open:

> *Updated element <xmi:id>: EA ExceptionHandler is transformed to an CallBehaviorAction with and input pin.*
>
> *Updated element <xmi:id>: EA InterruptFlow was transformed to an ExceptionHandler.*



ExceptionHandler.

ObjectFlow

An ObjectFlow line whose ends are not attached to any of the following elements will be transformed into a ControlFlow.

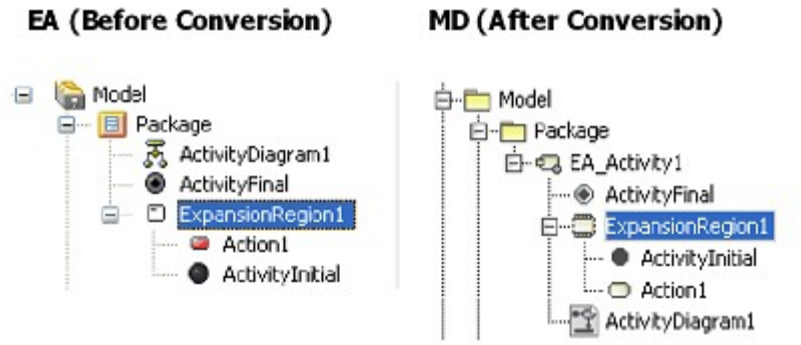
- InputPin
- OutputPin
- ObjectNode
- CentralBufferNode
- DataStoreNode

After completing the transformation, the following transformation message opens:

> *Updated element <xmi:id>: uml:ObjectFlow updated to uml:ControlFlow.*

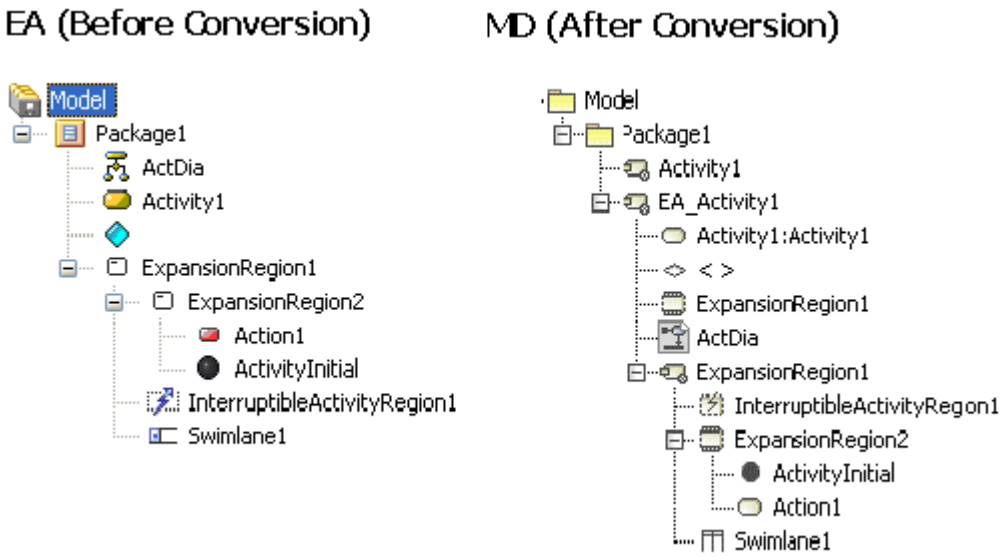
ExpansionRegion

Most of the elements placed inside any ExpansionRegion elements in EA will stay in their original place.



ExpansionRegion tree view.

However, if there is any Activity, Swimlane, InterruptibleActivityRegion, StructuredActivityNode, LooNode, SequenceNode, ConditionalNode, or other ExpansionRegion contained within an ExpansionRegion, it will be placed within a dummy Activity element. The created dummy will have the same name and will be placed at the same level as the ExpansionRegion element.



Nested ExpansionRegion tree view.

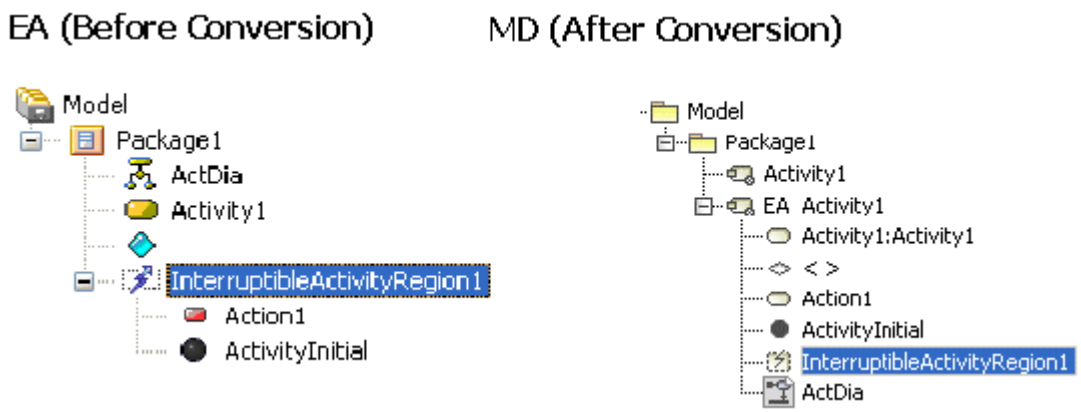
After completing the transformation, the following transformation message opens:

> *Updated element <xmi:id>: ExpansionRegion cannot contain some inner elements.*

> *An Activity with the same name as the ExpansionRegion was created to contain inner elements.*

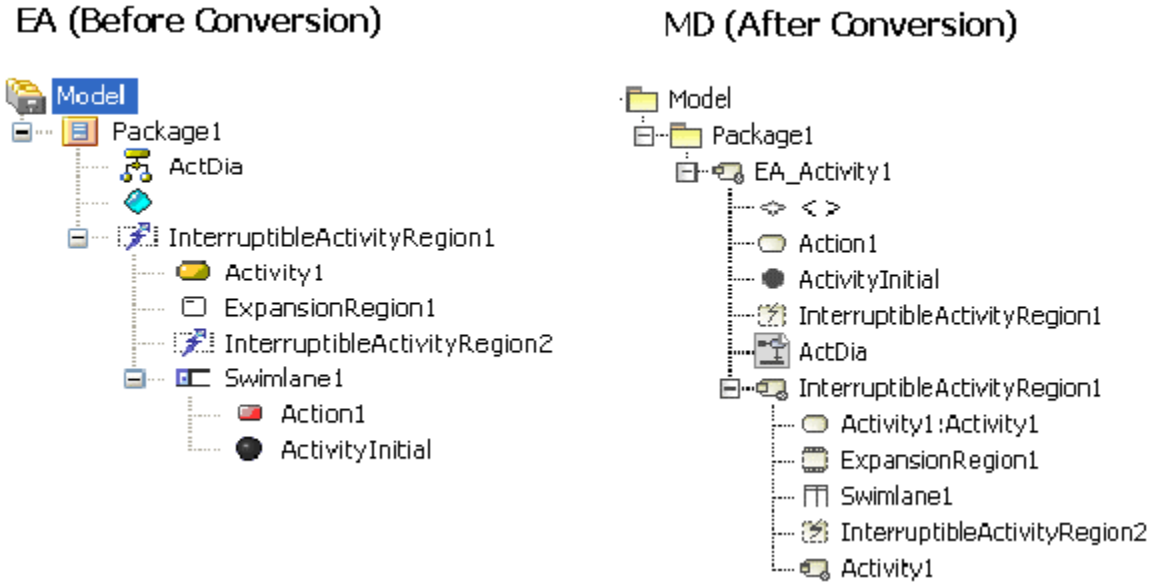
InterruptibleActivityRegion

Most of the elements placed inside an InterruptibleActivityRegion element in EA will be placed at the same level as the InterruptibleActivityRegion element.



InterruptibleActivityRegion tree view.

However, if there is any Activity, Swimlane, ExpansionRegion, StructuredActivityNode, LoopNode, SequenceNode, ConditionalNode, or other InterruptibleActivityRegion contained within an InterruptibleActivityRegion, it will be placed within a dummy Activity. The created dummy will have the same name and will be placed at the same level as the InterruptibleActivityRegion.



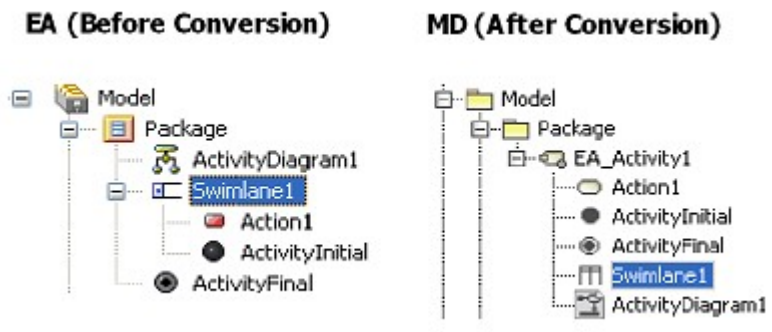
Nested InterruptibleActivityRegion tree view.

After completing the process, the following transformation message opens:

> *Updated element <xmi:id>: InterruptibleActivityRegion cannot contain some inner elements.*
>
> *An Activity with the same name as the InterruptibleActivityRegion was created to contain inner elements.*

Swimlane

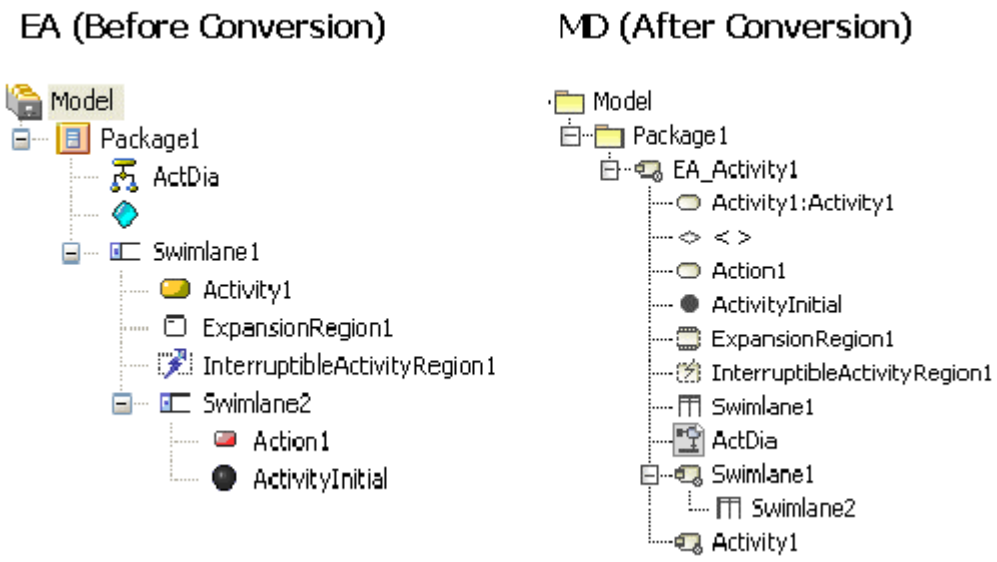
Most of the elements placed inside any Swimlane element in EA will be placed at the same level as the Swimlane element.



Swimlane tree view.

A dummy Activity will also be created to hold any other Swimlanes that it may contain. The dummy activity will have the same name and will be placed at the same level as the Swimlane.

If two or more Swimlanes are nested together, then every element (except Swimlane element) contained within either of them will be placed at the same level as the Swimlane topping the nested-Swimlane-elements hierarchy.



Nested Swimlane tree view.

After completing the process, the following transformation message opens:

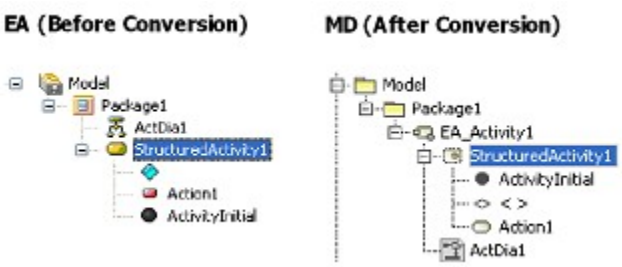
> *Updated element <xmi:id>: Swimlane cannot contain some inner elements. The XMI structure was fixed.*

StructuredActivity

Four elements are classified as Structured Activity elements in EA:

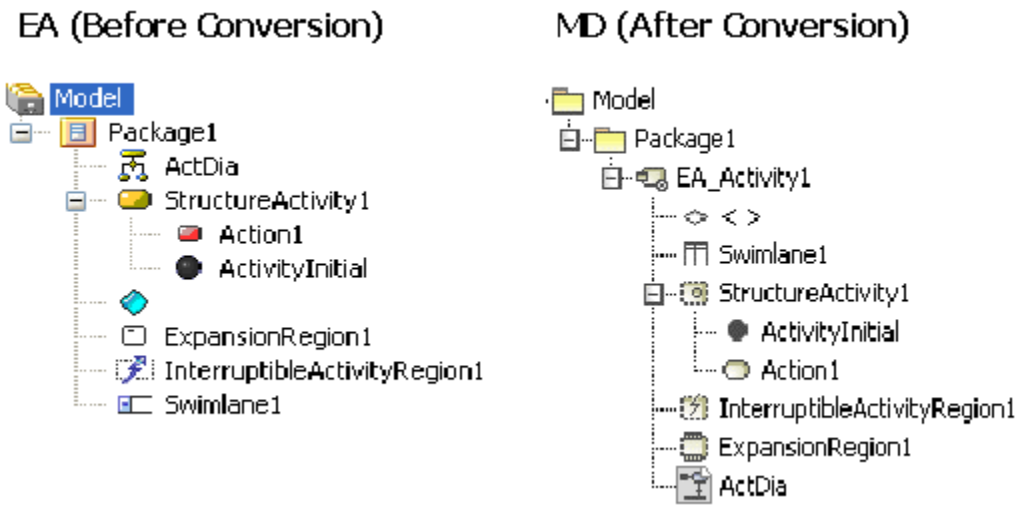
- StructuredActivityNode element
- LoopNode element
- SequenceNode element
- ConditionalNode element

Most of the elements placed inside any StructuredActivityNode, LoopNode, SequenceNode, or ConditionalNode elements in EA will stay in their original place.



StructuredActivityNode tree view.

However, if there is any Activity, Swimlane, InterruptibleActivityRegion, ExpansionRegion, or another StructuredActivity contained within a StructuredActivity, it will be placed within a dummy Activity. The created dummy will have the same name, and will be placed at the same level as the StructuredActivity.



Nested StructuredActivityNode tree view.

After completing the process, the following transformation messages will open, depending on the Structured Activity elements involved:

- Updated element <xmi:id>: StructuredActivityNode cannot contain some inner elements. An Activity with the same name as the StructuredActivityNode was created to contain inner elements.
- Updated element <xmi:id>: ConditionalNode cannot contain some inner elements. An Activity with the same name as the ConditionalNode was created to contain inner elements.
- Updated element <xmi:id>: LoopNode cannot contain some inner elements. An Activity with the same name as the LoopNode was created to contain inner elements.
- Updated element <xmi:id>: SequenceNode cannot contain some inner elements. An Activity with the same name as the SequenceNode was created to contain inner elements.

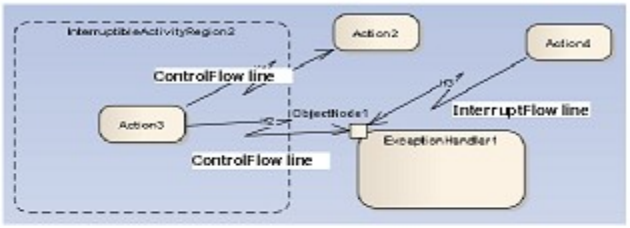
InterruptFlow

In some cases, EA InterruptFlows are ControlFlow lines. Their image will be displayed as the InterruptFlow line in the Activity diagram. An InterruptFlow is not a ControlFlow line if the InterruptFlow line is drawn from one element in an InterruptibleActivityRegion to another outside the InterruptibleActivityRegion. In an XMI file, this line will be imported as a ControlFlow line, and its image will be changed to that of the ControlFlow line.

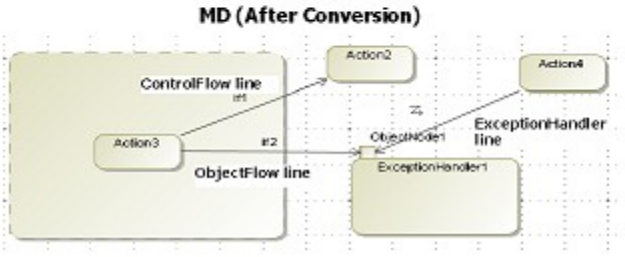
However, if either end of the line is any of the following elements, it will be transformed into an ObjectFlow line.

- InputPin element
- OutputPin element
- ObjectNode element
- CentralBufferNode element
- DataStoreNode element

EA (Before Conversion)



MD (After Conversion)

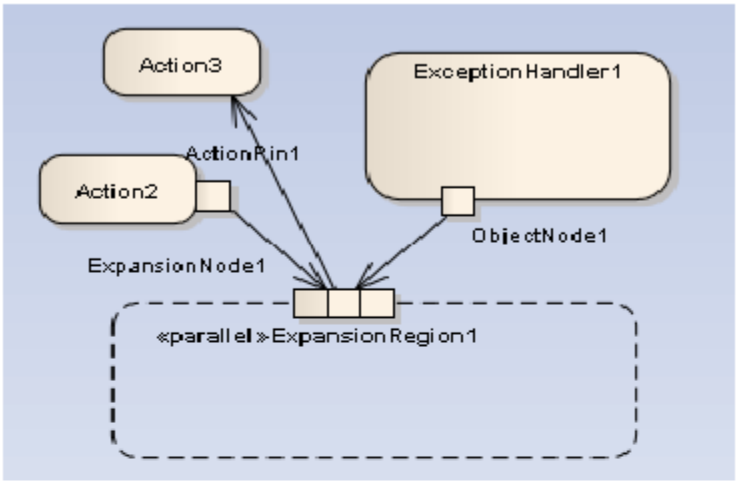


InterruptFlow.

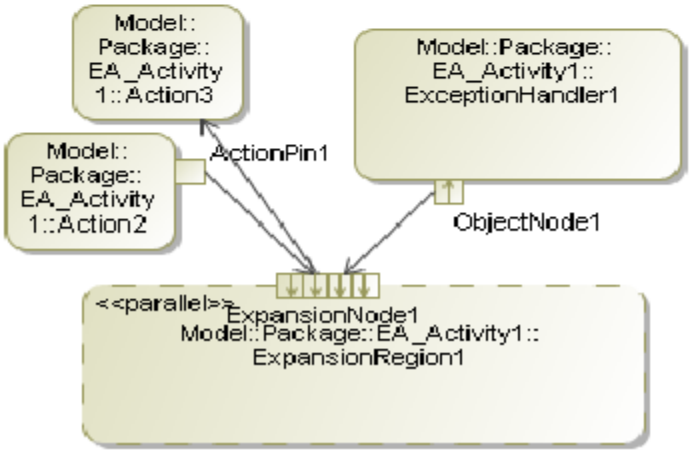
ExpansionNode

An ExpansionNode is a Pin which can only be contained within an ExpansionRegion and will be imported like any other Pin elements. However, if an ExpansionNode in EA is created inside another element rather than an ExpansionRegion, that particular ExpansionNode will not be imported.

EA (Before Conversion)



MD (After Conversion)



ExpansionNode.

Related pages

- Activity diagram