



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Composer Help

## Detaching Interactions

---

## Contents

- 1 Detaching Interactions
  - 1.1 Composer Level
  - 1.2 Project Level
  - 1.3 Block Level
  - 1.4 Inter-Session Communication

# Detaching Interactions

Starting with Release 8.1.400.35, Composer adds a new Project properties option, **Interaction Detach**, in the Orchestration Options dialog (shown below), which can generate the detach attribute in the `ixn:redirect` tag in the following Routing blocks: (**Default Route**, **Force Route**, **Queue Interaction**, **Route Interaction**, **Routing Rule**, and **Target**).

Selecting **Use Platform** in the dialog sets the attribute value to `true` and instructs Orchestration Server to detach an interaction from the current session before routing to the specified target, which can free Orchestration Server to start processing the next session. Keeping the default of **Use Application** causes the detach attribute not to be used, in which case the `<ixn:detach>` tag will be used by the generated SCXML application.

## Notes:

- Previously created Projects upgraded to this version will have the **Use Application** detach option and generate the `<ixn:detach>` tag for the detach operation. Also, the default value of the Detach property for the above Routing blocks is changed to `true`.
- Any new Routing blocks added will have the Detach Property default set to `true`.
- Diagrams upgraded to this version will continue to have the previously set values for the Detach property.

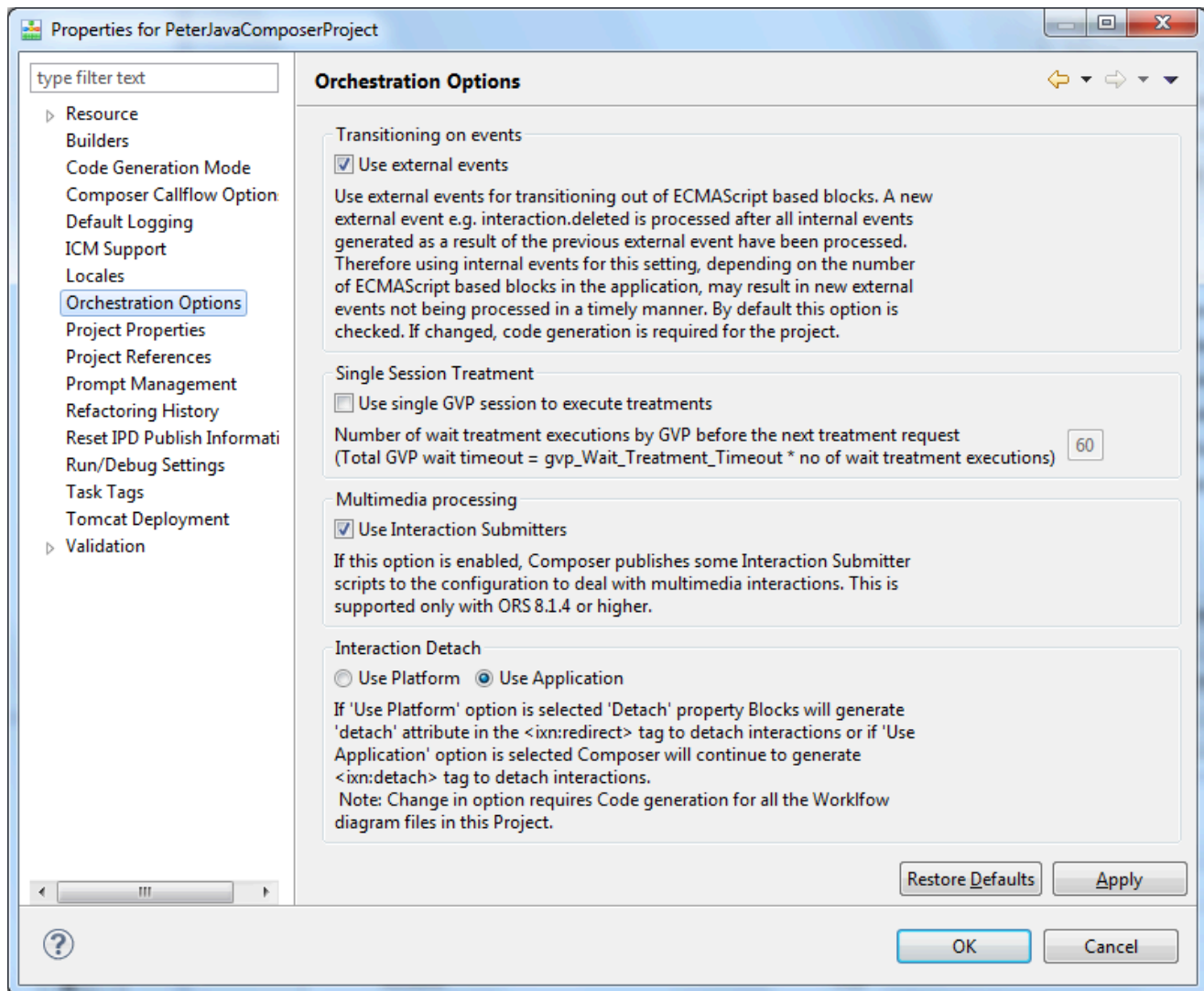
You can specify the detach method at the Composer level, at the Project level, or at the Block level.

## Composer Level

Use the **Detach** and **Attach** blocks available from the Flow Control palette group.

## Project Level

Right-click a Project, select **Properties > Orchestration Options**. The dialog box below opens.



Under **Interaction Detach**, the options are **Use Platform** or **Use Application**.

## [+] Use Platform

**Interaction Detach = Use Platform** (default value for new Projects)

If **Use Platform** is selected, Composer will generate the new detach attribute in the `<ixn:redirect>` tag in the above Routing blocks that have the Detach property enabled.

```
<state id="_reserved_Target1_redirect">
  <onentry>
    <ixn:redirect detach="true" requestid="App_Target1['requestid']"
interactionid="system.InteractionID" from="system.ThisDN"
    to="App_Target1['targetResource']" type="_genesys.queue.rType.RouteTypeDefault"
hints="_data.All_Locations"/>
  </onentry>
  <transition event="interaction.redirect.done">
```

```
cond="_event.data.requestid==App_Target1['requestid']" target=
  "$$_MY_PREFIX_$_reserved_NextBlock"/>
</state>
```

## eServices Blocks and Use Platform

**Use Platform** in SCXML code generation is applicable only for blocks that use the `<ixn:redirect>` tag. The following eServices blocks do not use `<ixn:redirect>` tag and therefore cannot use the Use Platform option shown above: Chat Transcript, Create Email, Email Forward, Email Response, and Create SMS. You can still change the default value for the Detach property in these blocks.

## Event Handling

When enabling the **Use Platform** option, Composer-generated code for the Routing blocks with the Detach property enabled will no longer handle detaching (`<ixn:detach>`) the interaction and re-attaching (`<ixn:attach>`) the interaction after a failed `<ixn:redirect>`.

The following events will no longer be handled by the generated code for these blocks:  
`interaction.detach.done` and `interaction.attach.done`.

Starting with 8144, Composer validates `error.interaction.attach` and `error.interaction.detach` exceptions in the Detachable blocks. If **Platform Detach** is enabled, the Detachable blocks having the above said exceptions will throw a warning.

The `interaction.deleted` event will be discarded at the application level (interaction process dialog Events) when its `resultof` attribute value is detaching in order to prevent premature application termination. As a result, application developers are no longer required to handle the `interaction.deleted` event in multiple places when detaching interactions.

## [+] Use Application

**Interaction Detach = Use Application** (default value for upgraded Projects)

If **Use Application** is selected, Composer will continue to generate the `<ixn:detach>` tag to detach interactions, in addition to the `<ixn:redirect>` tag.

```
<state id="_reserved_Target1_detach_run">
  <onentry>
    <ixn:detach interactionid="system.InteractionID"
requestid="App_Target1['requestid']"/>
  </onentry>
  <transition event="interaction.detach.done"
cond="_event.data.requestid==App_Target1['requestid']"
target="$$_MY_PREFIX_$_reserved_Target1_redirect"/>
</state>
<state id="_reserved_Target1_redirect">
  <onentry>
    <!-- NO detach attribute attribute usage. This code must run on ORS without in-
redirect detach support -->
    <ixn:redirect requestid="App_Target1['requestid']"
interactionid="system.InteractionID" from="system.ThisDN" to="App_Target1['targetResource']"
type="_genesys.queue.rType.RouteTypeDefault" hints="_data.All_Locations"/>
  </onentry>
```

```
<transition event="interaction.redirect.done"
cond="_event.data.requestid==App_Target1['requestid'])"
target="$$_MY_PREFIX_$_reserved_NextBlock"/>
<transition event="error.interaction.detach"
cond="_event.data.requestid==App_Target1['requestid'])"
target="$$_MY_PREFIX_$_reserved_Target1_attach">
  <script>
    App_Target1['error.rethrow'] =
    {'name':_event.name,params: {'requestid':_event.data.requestid,'error':_event.data.error,'description':_event.da
  </script>
</transition>
</state>
<state id="_reserved_Target1_attach">
  <onentry>
    <ixn:attach interactionid="system.InteractionID"
requestid="App_Target1['requestid']"/>
  </onentry>
  <transition event="interaction.attach.done"
cond="_event.data.requestid==App_Target1['requestid'])"
target="$$_MY_PREFIX_$_reserved_NextBlock">
    <!-- re-throw original error.interaction.redirect for the application -->
    <script>
      __Raise('error.interaction.redirect', App_Target1['error.rethrow'].params);
    </script>
  </transition>
</state>
```

## Block Level

If the Detach property is set to true for the above Routing blocks, Orchestration Server uses the new detach attribute and the interaction is detached from the session before routing to the specified target.

If the Detach property value is set to false (default), then no detach occurs before routing.

If the Target block Route property = true, then code generation will detach the interaction before queue:submit since the Orchestration Platform will redirect after queue:submit. This is applicable for both **Use Platform** and **Use Application** under **Interaction Detach**.

## Inter-Session Communication

Composer-generated applications support passing context from the source application session to the destination application session when the interaction is associated with a new Orchestration session after a detaching operation in the originating Orchestration session. This allows the new application that runs in the new session to access data (variables) previously collected in the originating session.

This feature is controlled by properties Pass Context, Pass Context Timeout of detachable blocks in the originating Orchestration session, and by properties Read Context, Read Context Timeout of the destination interaction process diagram (IPD).

To support this feature, the Composer originating application writes some data to the interaction user data just before detaching the interaction. When enabling the **Use Platform** option, this user data

update must be done right before the `<ixn:redirect>`.