



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

### Handling eServices Switchovers

---

## Contents

- 1 Handling eServices Switchovers
  - 1.1 Composer 8.1.400.19 and Earlier
  - 1.2 Between Composer 8.1.400.20 and 8.1.400.32
  - 1.3 Custom Events
  - 1.4 ORS resultof Attribute

# Handling eServices Switchovers

Starting with Composer 8.1.400.20, Composer default interaction process diagram (IPD) event handlers use the `interaction.present` event (named `interaction` re-queued in the IPD **Events** dialog) to handle eServices server switchover scenarios, such as a **Chat Sever HA switchover**. The default behavior was changed to wait for another `interaction.present` event indicating that the interaction is again available to the current session for processing.

## Composer 8.1.400.19 and Earlier

In previous versions of Composer (8.1.400.19 and earlier), the default behavior was to end the current session upon receiving `interaction.notcontrolled`. In such cases, upgrading to a newer version of Composer (8.1.400.33 or higher) will not replace this handler in existing IPDs even though it will add the handler for new IPDs.

## Between Composer 8.1.400.20 and 8.1.400.32

To handle switchovers in versions between Composer 8.1.400.20 and 8.1.400.32, add the `interaction` re-queued event in your IPD **Events** dialog and then **Generate Code**. If upgrading to version 8.1.400.33 or higher, your event handlers are automatically upgraded to a "custom" set of handlers and your old events will be retained. For more information, refer to **Intra Version Upgrades**.

## Custom Events

If you are using a custom event (for example, `interaction.notcontrolled`) to handle eServices/Interaction Server switchover or failover scenarios, please evaluate switching to the default event handler in the **Multimedia** category set of event handlers in the **Events** dialog.

## ORS resultof Attribute

If you decide to stay with `interaction.notcontrolled`, then Orchestration Server Release 8.1.400.17 offers a newly added `resultof` attribute to the `interaction.notcontrolled` event in your event handler, which can be used to make the event handler more specific.

Examples:

```
_event.data.interactionid == system.InteractionID && (!_event.data.resultof ||  
_event.data.resultof == 'deletion')
```

`resultof` attribute descriptions:

`detaching` - interaction became notcontrolled because it was detached from a running session.

`deletion` - interaction became notcontrolled because it was deleted.

`routing` - interaction became notcontrolled because of ORS actions such as delivering to an agent, placing in a queue, etc. So an event with `resultof= routing` is an expected event in a session because it is a result of some actions of the strategy itself.

`revoking` - interaction being processed by ORS (pulled) is taken away from that ORS instance explicitly (by agent, by media server, by Interaction Server due to inactivity) or implicitly (disconnect from Interaction Server). In these cases, all previously pulled interactions are no longer considered as such. So an event with `resultof= revoking` is a kind of unexpected event in a session. Ideally, a session should be terminated upon receiving such event, but this is up to application developer.