



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.

Orchestration Server Developer's Guide

[Detach an Interaction](#)

12/15/2025

Detach an Interaction

The following SCXML strategy detaches a call from the current session, then redirects it from DN 3001 on Switch 'tel_sw' to DN 7778 on Switch 'tel_sw_1'.

```
<!-- ***** -->
<!-- * TEST OF DETACH AND SUBSEQUENT REDIRECT * -->
<!-- ***** -->
<scxml version="1.0" xmlns="http://www.w3.org/2005/07/scxml"
  name="simple monitoring app"
  xmlns:queue="http://www.genesyslab.com/modules/queue"
  xmlns:ixn="http://www.genesyslab.com/modules/interaction"
  xmlns:dialog="http://www.genesyslab.com/modules/dialog" >
  <!-- ***** -->
  <datamodel>
    <data ID="stateName" expr="'initial'" />
    <data ID="reqid" expr="''" />
    <data ID="ixnid" expr="''" />
    <data ID="test_result" expr="'PASSED'" />
  </datamodel>
  <!-- ***** -->
  <initial>
    <transition target="waiting"/>
  </initial>
  <!-- ***** -->
  <state id="main" initial="waiting">
    <!-- ***** -->
    <state id="waiting">
      <onentry>
        <log expr="'Waiting for incoming call...'" />
      </onentry>
      <transition event="interaction.added" target="detachcall">
        <script>
          _data.ixnid = _event.data.interactionid;
          var dev =
            _genesys.ixn.interactions[_data.ixnid].parties[_data.ixnid + '-1'].device;
        </script>
        <log expr="'G_G_GOT initial interaction added: ' + uneval(
          _event.data )"/>
        <log expr="'_genesys.ixn: ' + uneval( _genesys.ixn )" />
      </transition>
    </state>
    <!-- ***** -->
    <!-- Detaching call from session with graceful termination by timeout -->
    <!-- ***** -->
    <state id="detachcall">
      <onentry>
        <script>
          var noMoreDetach = false;
          var detachTimeout = '3s'; /*'infinite' means endless
trying to detach*/
        </script>
        <if cond="detachTimeout != 'infinite'">
          <send event="'DetachTimeoutEvent'"
delay="detachTimeout"/>
        </if>
        <ixn:detach interactionid="_data.ixnid"
requestid="_data.reqid" />
      </onentry>
```

```

        <transition event="DetachTimeoutEvent">
            <script>
                noMoreDetach = true;
            </script>
        </transition>
        <transition event="error.interaction.detach" cond="(_event.data.error
== 'invalidstate')&&!noMoreDetach&&(_event.data.requestid == _data.reqid)">
            <log expr="'G_G_GOT detach invalidstate error: ' + uneval(
_event.data )" />
            <ixn:detach interactionid="_data.ixnid"
requestid="_data.reqid" />
        </transition>
        <transition event="error.interaction.detach" cond="(_event.data.error
== 'invalidstate')&&noMoreDetach&&(_event.data.requestid == _data.reqid)"
target="abnormal_exit">
            <log expr="'G_G_GOT final detach invalidstate error: ' +
uneval( _event.data )" />
        </transition>
        <transition event="error.interaction.detach"
cond="(_event.data.error != 'invalidstate')&&(_event.data.requestid == _data.reqid)"
target="abnormal_exit">
            <log expr="'G_G_GOT detach error: ' + uneval( _event.data )"
/>
        </transition>
        <transition event="interaction.detach.done"
cond="_event.data.requestid == _data.reqid" target="redirectcall">
            <log expr="'G_G_GOT detach done: ' + uneval( _event.data )"/>
        </transition>
        <!--*** Targetless transition to intercept interaction.deleted event
***-->
        <transition event="interaction.deleted"
cond="_event.data.interactionid == _data.ixnid">
            <log expr="'G_G_GOT detached interaction deleted: ' + uneval(
_event.data )" />
            <log expr="'_genesys.ixn: ' + uneval( _genesys.ixn )" />
        </transition>
    </state>
    <!-- *****-->
    <state id="redirectcall">
        <onentry>
            <ixn:redirect interactionid="_data.ixnid"
from="({'dn':'3001', 'switch':'tel_sw'})" to="({'dn':'7778', 'switch':'tel_sw_1'})"
requestid="_data.reqid" />
        </onentry>
        <transition event="error.interaction.redirect"
cond="_event.data.requestid == _data.reqid" target="abnormal_exit">
            <log expr="'G_G_GOT redirect error: ' + uneval( _event.data
)" />
        </transition>
        <transition event="interaction.redirect.done"
cond="_event.data.requestid == _data.reqid" target="somedelay">
            <log expr="'G_G_GOT redirect done: ' + uneval( _event.data
)"/>
        </transition>
    </state>
    <!-- *****-->
    <!--*** Just to make old and new sessions coexist after ixn_detach
(additional detach verification) ***-->
    <state id="somedelay">
        <onentry>
            <send event="'SynchroEvent'" delay="'33s'"/>
        </onentry>
        <transition event="SynchroEvent" target="exit"/>

```

```
        </state>
        <!--*** Common event handler for primary ixn deletion (just to terminate this
test session) ***-->
        <transition event="interaction.deleted" cond="_event.data.interactionid ==
_data.ixnid" target="exit">
            <log expr="'G_G_GOT initial interaction deleted: ' + uneval(
_event.data )"/>
            <log expr="'_genesys.ixn: ' + uneval( _genesys.ixn )" />
        </transition>
    </state>
    <!--_*****_-->
    <state id="abnormal_exit">
        <onentry>
            <log expr="'G_G_GOT ABNORMAL SESSION TERMINATION'" />
            <script>
                //Overwrite _data.test_result initial value that was set in
data block
                _data.test_result = 'FAILED';
            </script>
        </onentry>
        <transition target="exit"/>
    </state>
    <!--_*****_-->
    <final id="exit">
        <onentry>
            <log expr="'G_G_GOT SESSION TERMINATION WITH TEST_RESULT = ' +
_data.test_result" />
        </onentry>
    </final>
</scxml>
```