



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

Single Session Treatments

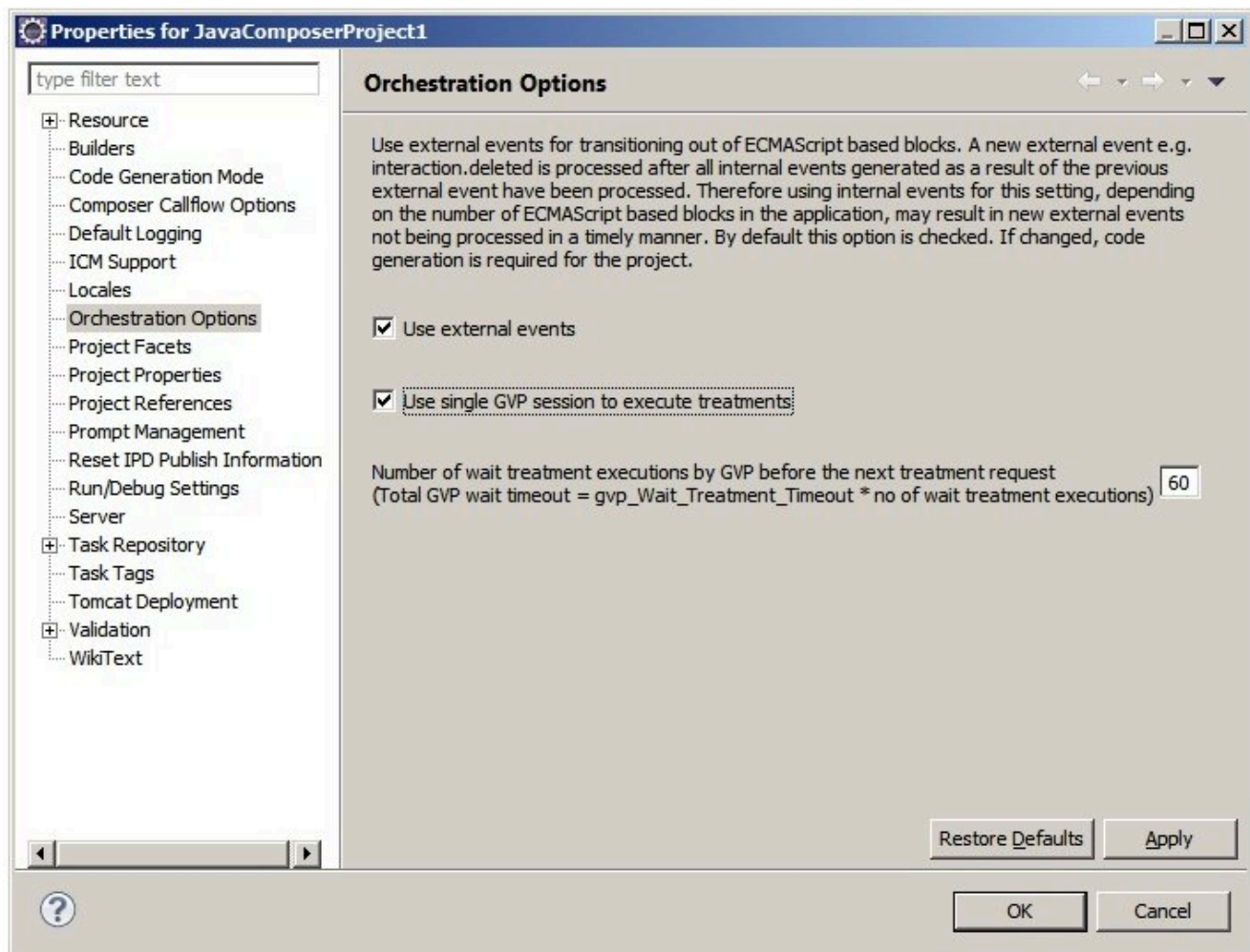
Contents

- 1 Single Session Treatments
 - 1.1 Workflow Block Changes
 - 1.2 IPD Diagram Changes
 - 1.3 Play Application Parameter Passing
 - 1.4 Frequently Asked Questions

Single Session Treatments

Starting with Composer 8.1.300.78, Composer supports ORS-centric GVP single-session treatments. When using the Play Application, Play Sound (Music and ARM Types) Exit, and Disconnect blocks, voice applications can now optionally use a single VXML session on Media Control Platform to play/run multiple treatments instead of using one session per treatment. This enables DTMF buffering between multiple MSML treatments.

A new static master VXML Page is used for playing treatments in one GVP session: (e.g., `ComposerPlayTreatments.vxml`). Treatment URLs to play can be Composer diagram-generated vxml file, MCP built in treatments or Third party sub dialog URLs. The `ORSSessionID` will be provided as initial Play Application Input parameters; successive HTTP requests can pass treatment input parameters. Treatment execution results and return results will be passed back to the ORS session. The Orchestration Options dialog box is used for this functionality.



Workflow Block Changes

- Main workflow -> Exit /Disconnect Block: Generates code to end the GVP session if active based on `KeepGVPSessionAlive` and `isGVPSessionActive` flags.
- Play Application / Play Sound (music) -> Change in code to simulate treatments.

IPD Diagram Changes

- New event handlers have been added to handle communication between the GVP master page and the SCXML application.
1. `composer.dialog.play` - Play Application block request to IPD diagram for treatment request after the Master VXML page is in action.
 2. `composer.dialog.start.done` - Master vxml page request to ORS session for treatment done.
 3. `error.composer.dialog.start` - Master vxml page request to ORS session for treatment error.
 4. `composer.dialog.exit` - Exit/Disconnect block request to IPD diagram for exiting GVP session.
 5. `composer.workflow.proceed` - IPD diagram to Play Application block event to proceed further (compare to `dialog.start.done` or `dialog.start.requestid`).
 6. `composer.workflow.exit` - IPD diagram response event to `composer.dialog.exit` request from Exit or Disconnect block event to proceed exit (After GVP session exists response "this event" is thrown).

Play Application Parameter Passing

In the traditional ORS treatments model, Play application treatment parameter passing happens via User Data. which has been sent along with the treatment request to GVP. In the Single GVP session model, User Data availability will be a limitation as the successive treatment requests happens via events. However, there are no changes to the input/output parameters defining mechanism in the Play Application block. That means the input/output parameters in Play Application blocks will be sent to the master VXML via events and then be propagated to each subdialog in `system_treatment_params` json object.

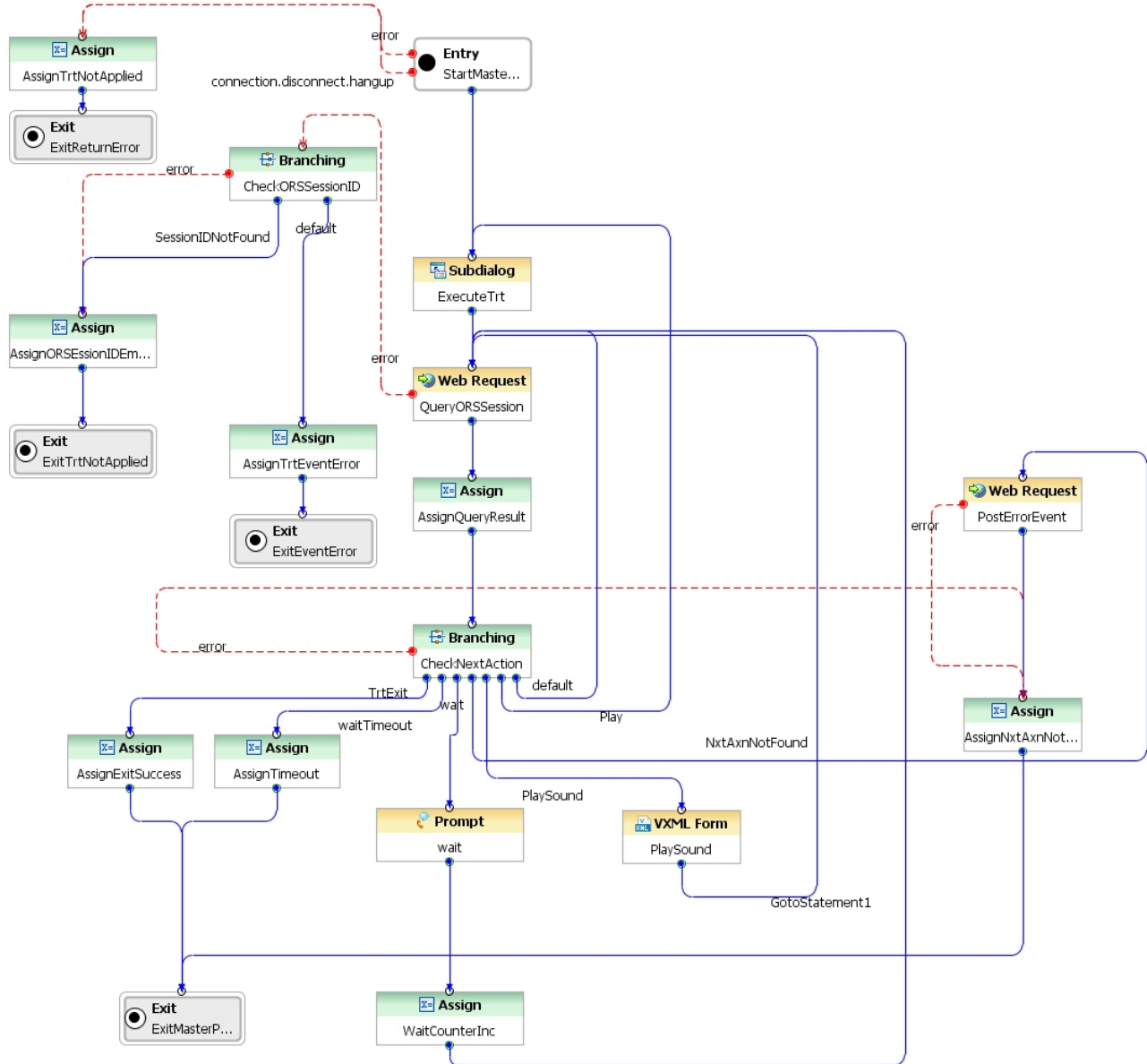
To receive Input parameters:

- In sub-callflows, instead of using `InteractionData` blocks to access userdata, use the `system_treatment_params` object's properties. First `<form>` in the Sub-callflow should have the following lines to receive the input parameters and the treatment language:

Note: Even if the Play Application block doesn't send or receive parameters, *the above `<var>` tags must be added to the Sub-Callflows since the Master VXML always sends these parameters.*

To Return Output Parameters to Play Application blocks: No changes here; variables defined in the Exit block will return to the Play Application block (if `Wait for Treatment End` is set to true).

Master VXML Flow Page:



GVP Wait and Session Timeouts:

The GVP Master VXML application can wait to receive the next treatment request from the ORS session. The GVP wait timeout before requesting ORS is now controlled on the ORS side instead of GVP side due to Prompt queuing impacts. Following are the options available to control GVP wait timeout between treatments and GVP session timeout:

1. gvp_Wait_Treatment_Timeout variable on the IPD diagram file will be used to delay responses to GVP when no treatment is ready to be executed. The Default value of this variable is 1s and this can be configured at the ERS object -> Application Params section in Configuration Manager. If configured the new value will take effect.

2. Orchestration Option -> Property page -> Number of wait treatment requests should GVP wait to end the session. When there is no treatment to play for a while, this option is used to control the GVP session timeout.
3. Total GVP session timeout = Number of wait requests * gvp_Wait_Treatment_Timeout. For example, 10 continuous wait requests with a delay of 3 seconds will provide 30-seconds of GVP Session timeout for a idle case -> $10 * 3s = 30 \text{ seconds}$.

Frequently Asked Questions

How can I run treatments in a separate GVP session while not disturbing the Master VXML session?
Using Assign blocks, set flag `_data.keepGVPSessionAlive` to false before your treatment block and reset to true after your treatment block.

What if ORS logic does not have any treatments after the first Play Application treatment?
The Master VXML session will poll ORS (IPD Diagram SCXML) based on the `gvp_Wait_Treatment_Timeout` value until it reaches the maximum GVP Session Timeout (configurable via Composer Project properties) and ends the session as GVP Session timeout.

What if the workflow wants to end the Master VXML session in the middle of a call?
The workflow may have to send the 'exit' treatment request to GVP to end the session.

How do I send and receive parameters in this GVP Single Session model?
Passing input parameters to the GVP session will continue to work using the Udata methodology with no changes. Receiving output parameters will happen via `composer.dialog.start.done` event (`composer_treatment_result` option).

What if the "GVP Keepalive" option is unchecked in the Orchestration Options property page?
Code generation is required whenever there are changes in the Orchestration Options properties page. If the "GVP Keepalive" option is unchecked, code generation will generate code for normal ORS-MCP paradigm treatments.

How do I end the Master VXML treatment session in a workflow diagram?
Use SCXML state block to generate an "exit" request to the GVP Master Session.

What if the treatment URL is wrong or not available?
The Master VXML will try to execute the treatment URL as subdialog and return back the return results. If the URL cannot be executed, `error.composer.dialog.start` will be returned back as error result to IPD diagram.

How do I cancel the Master VXML treatment dialog using ORS Cancel Call?
The Master VXML session's dialog requestID will be available in the `_data.gvpMasterTreatmentRequestID` variable of the application. To manually end the GVP master VXML session use the following code snippet in a SCXML State block.

```
<state id="CancelMasterVXML">
  <onentry>
    <script>
```

```
                var composerRequestID = _sessionId+'_'+generateComposerTrtRequestID();
            </script>
<if cond="_data.keepGVPSessionAlive==true &&_data.isGVPSessionActive==true">
    <send event="'composer.dialog.exit'">
        <param name="treatment_id" expr="'exit'"/>
        <param name="treatment_url" expr="''"/>
        <param name="treatment_GVPRequestID"
expr="composerRequestID"/>
        <param name="treatment_waitForEnd" expr="'true'"/>
    </send>
    <else/>
        <raise event="'proceedNext'"/>
    </if>
</onentry>
<transition event="composer.workflow.exit" cond="_event.data.requestID ==
composerRequestID"
    target="$NextBlock$">
<script>
        _data.isGVPSessionActive =false;
        _data.keepGVPSessionAlive = false;           // Optional - Set
keepGVPSessionActive to false
        to completely end the Single GVP session and run ordinary ORS treatments
onwards.
</script>
    </transition>
    <transition event="proceedNext" target="$NextBlock$">
    </transition>
```

What if the treatment URL VXML page is a Main type callflow (does not have the <return> tag, but instead the <exit> tag)?

If a Main type callflow diagram is used, a treatment VXML page would have an <exit> tag, which will end the GVP session and return back to ORS. As a result, successive treatment requests will start a new Master VXML page. For this reason, Sub callflows are required to maintain a Master VXML page session.

When an ORS/workflow diagram does not have any treatment to execute, what will GVP do and how often will it poll an ORS session?

GVP will execute "wait" (silence) treatments while waiting for the next treatment request from ORS. GVP polls ORS based on the gvp_Wait_Treatment_Timeout value until the GVP session timeout is reached.