



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

Play Application Block

# Play Application Block

## Contents

- **1 Play Application Block**
  - 1.1 Name Property
  - 1.2 Block Notes Property
  - 1.3 Exceptions Property
  - 1.4 Condition Property
  - 1.5 Logging Details Property
  - 1.6 Log Level Property
  - 1.7 Enable Status Property
  - 1.8 Device ID Property
  - 1.9 Hints Property
  - 1.10 Interaction ID Property
  - 1.11 Extensions Property
  - 1.12 Language Property
  - 1.13 Parameters Property
  - 1.14 Resource Property
  - 1.15 Type Property
  - 1.16 Use User Data Property
  - 1.17 Request ID Property
  - 1.18 Wait For Treatment End Property
  - 1.19 ORS Extensions Property
  - 1.20 Passing Parameters
  - 1.21 Single Session Treatments

Use to execute an application (such as a Composer voice application) or a script on a device, such as an IVR.

- For URS-controlled applications where the call lands on the strategy first and then a VXML application is called using the Play Application treatment, **GVP** must be configured as DN type VoiceOverIP Service. Find detailed configuration instructions in Voice Platform Solution Integration Guide.
- **Debugging** a Play Application block will not step into the associated Callflow diagram and will not launch a GVP debugging sessions. Instead debugging will continue on to the block after the Play Application block.

Also see **Passing Parameters**. The Play Application block has the following properties:

### Name Property

Find this property's details under **Common Properties**.

### Block Notes Property

Find this property's details under **Common Properties**.

### Exceptions Property

Find this property's details under **Common Properties**.

### Condition Property

Find this property's details under **Common Properties**.

### Logging Details Property

Find this property's details under **Common Properties**.

### Log Level Property

Find this property's details under **Common Properties**.

## Enable Status Property

Find this property's details under [Common Properties](#).

## Device ID Property

If specified, ORS will play treatments itself; otherwise, treatment playing is delegated to URS. The device should specify the DN where the call is currently located. If the call is on multiple DNs, specify the DN for which the treatment will be applied. Users can enter a value or select any runtime variable from the dropdown.

## Hints Property

This property is for future use by Orchestration Server. Its use will be described in various action elements reference in the Orchestration Server wiki.

## Interaction ID Property

Set to a meaningful value or keep the default value, which is the system variable `InteractionId`. Can be used for "interaction-less" processing for scenarios where the `InteractionId` variable is not automatically initialized, but instead must wait for an event. An example would be an SCXML application triggered by a Web Service that does not add an interaction. Background: Previous to 8.1.1, Composer did not expose an Interaction ID property. Instead, when ORS started processing an interaction, a generated SCXML application automatically initialized the system variable, `InteractionId`. This variable was then used internally by Routing and certain eServices blocks when interacting with ORS. With the introduction of support for Interaction-less processing, you can now define a specific event ([IPD Wait For Event](#) property) to initialize `InteractionId`, or not define an event at all. For scenarios with an interaction ([IPD Diagram/Wait For Event=interaction.present](#) for example), you may keep the default value for the Interaction ID property. The default value is the system variable `InteractionId`, which is initialized automatically in this case. For other scenarios (any scenario where the system variable `InteractionId` is not set), you may choose to:

1. Not use blocks that require an Interaction ID
2. And/or set the Interaction ID property to a meaningful value
3. And/or assign a meaningful value to the `InteractionId` system variable

## Extensions Property

Select the variable to retrieve extensions data in event `dialog.start.done` as described in the *Orchestration Developers Guide*, Orchestration Extensions, Dialog Log Interface section of the [Orchestration Server Wiki](#).

## Language Property


To set the active language:

1. Select the Language row in the block's property table.
2. Click under Value to display a down arrow.
3. Select one of the following languages:

- **English (US)**
- **Spanish**
- **Mandarin**
- **Cantonese**
- **Vietnamese**
- **French**
- **French (Canada)**
- **German**
- **Italian**
- **Japanese**
- **Korean**
- **Russian**

## Parameters Property

If the **Type** property is not URL, use the Parameters property to specify any treatment parameters:

1. Click the **Parameters** row in the block's property table.
2. Click the  button to open the Input and Output Parameter Sync dialog box.
3. Click **Add** to add a treatment parameter.
4. Under **Parameter Name**, accept the default name or change it.
5. From the **Parameter Type** drop-down list, select input.

Note: If the Type property is set to **ProjectFile**, the dialog box shows only input variables as Composer validates against the specified callflow.

6. Under **Value**, select from among the **variables** shown.
7. Under **Definition**, type a description for this parameter.
8. Click **Add** again to enter another parameter, or click **OK** to finish.

To delete a parameter, select an entry from the list and click **Delete**. Also see the [Passing Parameters](#) section below. **Next Generation Interpreter (NGI) Change** In 8.1 and prior releases, a parameter in the SIP Request URI was exposed as a VoiceXML session variable as follows:

- Request URI: sip:dialog@host;ParameterA=ValueA
- Session Variable: session.connection.protocol.sip.requesturi['ParameterA'] (Note: The case is preserved)


Starting from 8.1.1, the parameter name will be converted to lowercase in the session variable's array index. Example:

- Request URI: sip:dialog@host;ParameterA=ValueA
- Session Variable: session.connection.protocol.sip.requesturi['parametera'] (Note: The index has to be lowercase.)

Any parameters set in the Play Application block will be accessed in NGI with the parameter name in all lowercase.

## Resource Property

Use to specify the treatment resource.

1. Click under **Value**.
2. The Resource property works with the Type property.
  - If **URL** or **Id** is the selected Type, clicking under Value displays a down arrow. Click the down arrow and select a [variable](#). For GVP applications, the variable must contain the full URI of the VXML page.
  - If **ProjectFile** is the selected Type, clicking under Value displays the  button. Click to open a dialog box where you can select the Project file/callflow for the treatment application. Integration with voice callflows is provided so it is possible to select a callflow diagram file as the project file. In this case, Composer will automatically substitute the actual URL of the page at the time of Code Generation.

## Type Property

Identifies the type of Resource.

1. Click under **Value** to display a down arrow.
2. Select one of the following types:
  - **URL**--Indicates that the Resource is a .vxml file and not a callflow. Composer currently does not parse this .vxml file so the Input and Output Parameter Sync dialog box described under the Parameters property does not open.
  - **ProjectFile**--Indicates that the Resource is a callflow.

- **Id**--Indicates the ScriptID parameter as described in the *Genesys Voice Platform 8.1 Reference Help* (see Sample VoiceXML Applications > CTI Interactions > Treatments (Post Initiate Transfer) > Play Application topic). The Play Application treatment is invoked without APP\_URL specified in the strategy.

## Use User Data Property

When set to true, Composer will automatically update the interaction's **User Data** with the input/inout parameters specified in the Parameters property. Similarly, Composer will automatically read the interaction's User Data and update corresponding variables for every inout/output parameter specified in the Parameters property. Uses to avoid the procedure described in the **Passing Parameters** property description.

## Request ID Property

Select the variable to hold the ID associated with the treatment request from the orchestration application or the resource.

## Wait For Treatment End Property

Select **true** or **false**.

- If true, the transition to the next block occurs when the treatment is finished (or if a timeout occurs).
- If set to false, processing goes to the next block once the treatment is successfully started instead of waiting for the treatment to complete. The Request ID variable holds the ID of the treatment.

## ORS Extensions Property

Starting with 8.1.4, Composer blocks used to build routing applications (with the exception of the Disconnect and EndParallel blocks) add a new **ORS Extensions** property.

## Passing Parameters

To pass parameters from a workflow to a callflow that is being invoked with the Play Application block, follow these steps: Note: Step 2. below is required only for SIPS and is not mandatory for CTIC flow cases.

1. Create a variable of type Input in the Variables property of the callflow's Entry block. The variable name must be all lowercase.

2. At a point before the Play Application block in the workflow, add an **ECMAScript** block with the following script contents:

`var input = new Object(); input.xyz = _data.VariableToPass; _genesys.ixn.setuData(input);` xyz should match the variable name that you created in step 1. You can replace `_data.VariableToPass` with any variable, or a literal value, such as a string or number.

3. In the callflow, access the value using the variable that was created in step 1. Its value will be set automatically to the value specified in the ECMAScript block at the beginning of the callflow. Its value will be set at the beginning of the callflow to the value specified in the ECMAScript block in the workflow.

To pass parameters from a callflow (invoked from a Play Application block) back to the workflow, follow these steps:

1. Create a variable of type User in the Variables property of the callflow's Entry block. The variable name must be all lowercase.
2. In the callflow, use this variable to store any value that you want to pass back to the workflow.
3. In the callflow's Exit block, specify the variable as a return value.
4. In the workflow, after the Play Application block, the value will be available in the workflow variable `_genesys.ixn.interactions[ixnid].udata.xyz`, where xyz matches the name of the variable created in step 1.

## 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. For more information, see [Single Session Treatments](#).