



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

Subroutine Block

# Subroutine Block

## Contents

- **1 Subroutine Block**
  - 1.1 Creating a Subroutine Using A Sub-Workflow
  - 1.2 Name Property
  - 1.3 Block Notes Property
  - 1.4 Exceptions Property
  - 1.5 Uri Property
  - 1.6 Type Property
  - 1.7 Condition Property
  - 1.8 Logging Details Property
  - 1.9 Log Level Property
  - 1.10 Enable Status Property
  - 1.11 Parameters Property
  - 1.12 ORS Extensions Property

Use the Subroutine block to create reusable sub-modules (sub-workflows). You can invoke external SCXML documents or use a sub-workflow created using Composer. The input and output parameters names will be automatically picked from the sub-workflow created by Composer. Composer supports passing variables between a workflow and sub-workflow.

Also see [Using Composer Shared Subroutines](#).

### Important

Starting with Composer 8.1.3 versions, the callflow diagram Subdialog block and the workflow diagram Subroutine block use absolute paths with the Project name to refer to the location of the selected resource in the Workspace, e.g., `workspace:///WFM/Workflows/subroutine.workflow`. Renaming or copying Projects requires a manual update to change the Project name in the Subroutine and Subdialog blocks.

## Creating a Subroutine Using A Sub-Workflow

1. Create the main workflow diagram file using **New > Other > Composer > Workflow diagram > Main Workflow**.
2. After designing the main workflow diagram, create the sub-workflow diagram using **New > Other > Composer > Workflow diagram > Sub-Workflow**.
3. In the Entry Block of the sub-workflow diagram, enter the parameters, which will be passed as input from the main to the sub-workflow diagram.

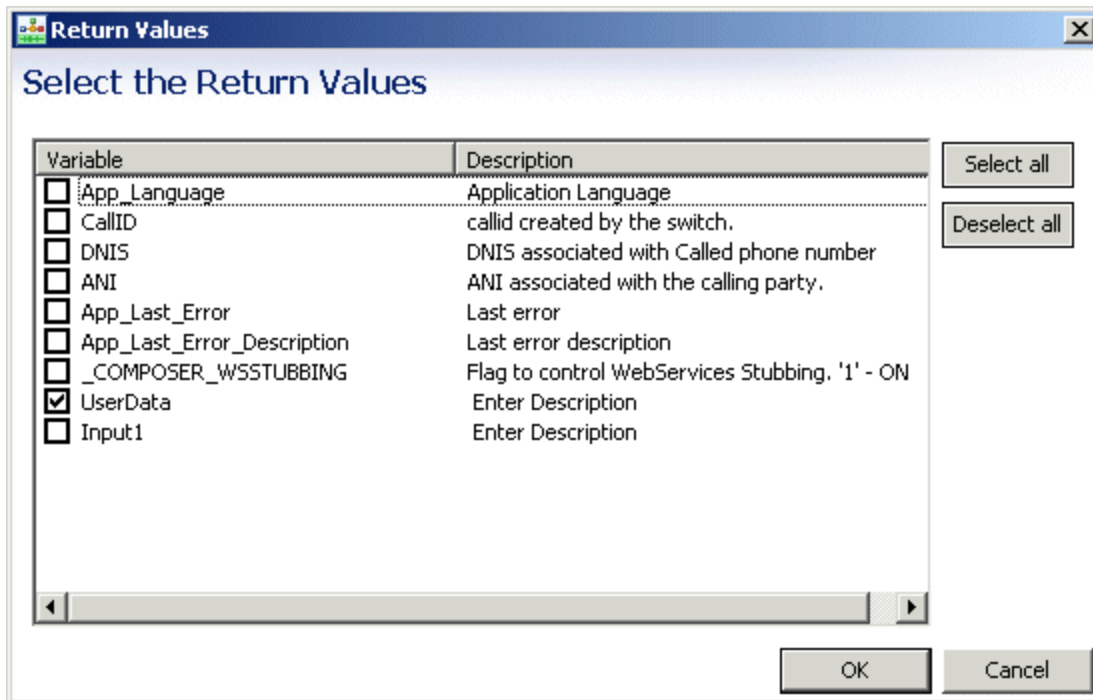
**Variable Settings**

**Application Variables**

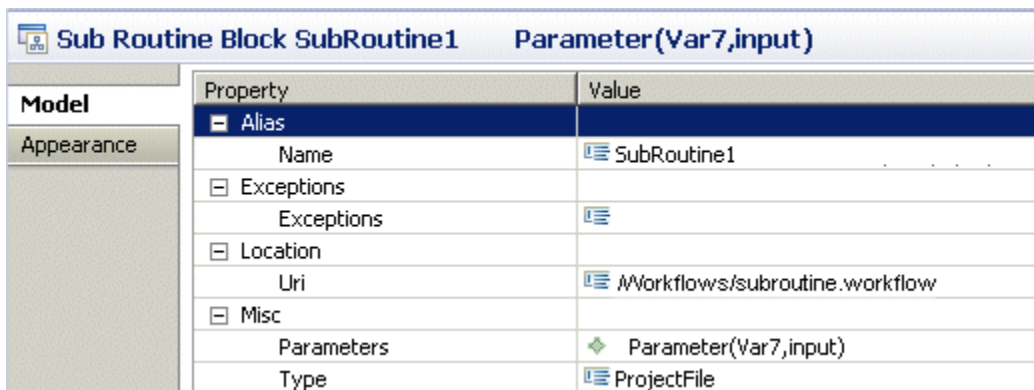
Variable Name	Type	Value	Description
App_Language	System	en-US	Application Language
CallID	System	_genesys.ixn.interactions[0].voice.callid	callid created by the switch
DNIS	System	_genesys.ixn.interactions[0].voice.dnis	DNIS associated with Call...
ANI	System	_genesys.ixn.interactions[0].voice.ani	ANI associated with the c...
App_Last_Error	System	undefined	Last error
App_Last_Error_Description	System	undefined	Last error description
_COMPOSER_WSSSTUBBING	User	0	Flag to control WebServic...
Input1	Input	Enter Value	Enter Description

Input dropdown menu options: User, Input

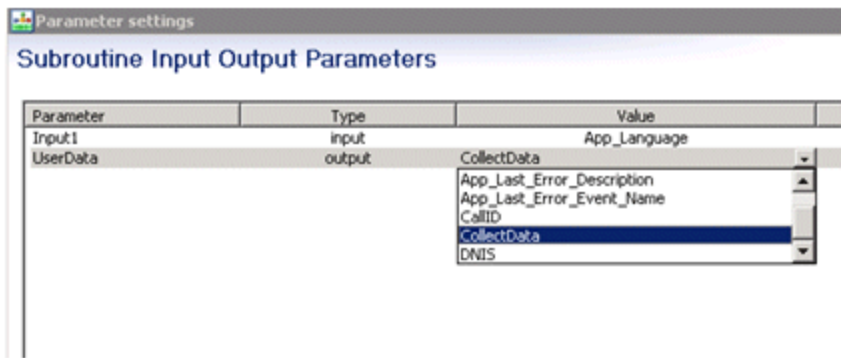
4. Design the sub-workflow diagram.
5. In the Exit block of the sub-workflow diagram, select the variables, which will be returned back to the called main diagram.



6. In the main diagram, use the Subroutine Block to call the newly created sub-workflow and the input and output parameters. For input/output synchronization, use the Uri property of the Subroutine block to select the sub-workflow diagram.



Now the Parameters property can be used for the Parameter Synchronization. The main diagram implicitly parses the sub-workflow parameters and lists them in the Parameter settings dialog as shown below.



7. Define the value for the input type variables and collect the returning output type variables in a variable.

The Subroutine 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](#).


### Uri Property

The Uri property specifies the destination (URL or Composer Project) depending on the value of the Type property. To set a URL destination for the Uri property (**Type** property is set to **URL**):

1. Select the **Uri** row in the block's property table.
2. In the **Value** field, type a valid URL. Variables should not be selected as all subroutines are fetched by Orchestration Server before it starts executing the application at which time variables do not exist.

To set a Project destination for the Uri property (**Type** property is set to ProjectFile):

1. Click the **Uri** row in the block's property table.

2. Click the  button to open the Uri dialog box.
3. Select a workflow in the list.
4. Click **OK** to close the dialog box.

## Type Property

The Type property sets the type of the invoked subroutine. There are two options:

- **URL**--The invoked sub-workflow can be found at the location specified in the Uri property.
- **ProjectFile**--The invoked sub-workflow is another workflow in the Project.

To select a value for the Type property:

1. Select the **Type** row in the block's property table.
2. In the **Value** field, select URL or ProjectFile from the drop-down list.

## 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](#).


## Parameters Property

Use the Parameters property to specify parameters to pass to the invoked sub-workflow. To specify

parameters: The URI field must contain a value.

1. Click the **Parameters** row under **Value**.
2. Click the  button to open the Subroutine Input Output Parameters dialog box.
3. Click the **Add** button to enter parameter details.
4. In the **Parameter** field, accept the default name or change it.
5. From the **Type** drop-down list, select input, output, or inout:

<b>input</b>	Input parameters are variables submitted to the sub-workflow.
<b>output</b>	Output parameters are variables that the sub-workflow returns and will be reassigned back to the current workflow.
<b>input/output</b>	Inout parameters are parameters that act as both input and output.

6. In the **Expression** drop-down list, select from among the variables shown, type your own expression, or click the  button to use **Expression Builder**.
7. In the **Description** field, type a description for this parameter.
8. Click **Add** again to enter another parameter, or click **OK** to finish.

## 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.