

# **GENESYS**

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.



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

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.

# 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.

#### 🎂 Yariable Settings

#### **Application Variables**

Variable Name	Туре	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_WSSTUBBING	User	0	Flag to control WebServic
Input1	Input 💌	Enter Value	Enter Description
	User	1	
	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.

ariable App_Language CallID DNIS ANI App_Last_Error App_Last_Error_Description _COMPOSER_WSSTUBBING UserData Input1	Description Application Language callid created by the switch. DNIS associated with Called phone number ANI associated with the calling party. Last error Last error East error description Flag to control WebServices Stubbing, '1' - ON Enter Description Enter Description	Deselect al

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.

Sub Routine Block SubRoutine1 Parameter(Var7,input)		
Model	Property Alias	Value
Appearance	Name	🖙 SubRoutine1
	Exceptions	
	Exceptions	E
	E Location	
	Uri	Workflows/subroutine.workflow
	🖂 Misc	
	Parameters	Parameter(Var7,input)
	Туре	ProjectFile

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.

Parameter	Туре	Value	
Inputi	input	App_Language	
UserData	output	CollectData	
		App_Last_Error_Description App_Last_Error_Event_Name CallID CollectData	-

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 (**Typ**e property is set to **UR**L):

- 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:

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

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

- 1. In the **Value** drop-down list, select from among the variables shown, type your own expression, or click the EME button to use Skill Expression Builder.
- 2. In the **Definition** field, type a description for this parameter.
- 3. Click **Add** again to enter another parameter, or click **OK** to finish.

Delete Button To delete a parameter:

- 1. Select an entry from the list.
- 2. Click Delete.