

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.

Composer Help

Route Flow Control Blocks

Route Flow Control Blocks

The table below summarizes the routing blocks used for flow control.

Assign Block Attach Block Attach Block Begin Parallel Block Begin Parallel Block Branching Block Use to enable the design of multiple threads, such as running busy treatments in parallel files. Use as a decision point in a callflow or workflow. It enables you to specify multiple application routes based on a branching condition. Depending on which condition is satisfied, the call follows the corresponding application route. Cancel Event Block Detach Block Use to cancel a custom event. Use for detaching a specific interaction from the current Orchestration Server session. Use to invoke the Cancel Call treatment, which ends the strategy and deletes the interaction from URS memory. ECMAScript Block End Parallel Block Loe to mark the end of the threads that were started by a matching Begin Parallel block. All routing strategy diagrams must start with an Entry block. Defines variables that can be shared across different blocks in the same workflow. The Entry block cannot have any incoming connections. Use to terminate the workflow or to return control back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. Looping Block Sends a response to a request-based event from an external application to the Orchestration Platform. SCXML State Block Use to invoke external SCXML document that it generates based on the workflow dicagram.		
Begin Parallel Block Begin Parallel Block Use to enable the design of multiple threads, such as running busy treatments in parallel files. Use as a decision point in a callflow or workflow. It enables you to specify multiple application routes based on a branching condition. Depending on which condition is satisfied, the call follows the corresponding application route. Cancel Event Block Detach Block Use to cancel a custom event. Detach Block Use for detaching a specific interaction from the current Orchestration Server session. Use to invoke the Cancel Call treatment, which ends the strategy and deletes the interaction from URS memory. ECMAScript Block End Parallel Block Use to build an ECMA Script expression for routing decisions. Lose to mark the end of the threads that were started by a matching Begin Parallel block. All routing strategy diagrams must start with an Entry block. Defines variables that can be shared across different blocks in the same workflow. The Entry block cannot have any incoming connections. Use to terminate the workflow or to return control back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. Looping Block Looping Block Use to throw custom events. Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Subravitine Block Use to invoke external SCXML documents or a sub-	Assign Block	
as running busy treatments in parallel files. Use as a decision point in a callflow or workflow. It enables you to specify multiple application routes based on a branching condition. Depending on which condition is satisfied, the call follows the corresponding application route. Cancel Event Block Detach Block Use to cancel a custom event. Use for detaching a specific interaction from the current Orchestration Server session. Use to invoke the Cancel Call treatment, which ends the strategy and deletes the interaction from URS memory. ECMAScript Block End Parallel Block Use to build an ECMA Script expression for routing decisions. Entry Block Use to mark the end of the threads that were started by a matching Begin Parallel block. All routing strategy diagrams must start with an Entry block. Defines variables that can be shared across different blocks in the same workflow. The Entry block cannot have any incoming connections. Use to terminate the workflow or to return control back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. Looping Block In cases where multiple records are returned, the Looping block can loop through all the records. Raise Event Block Sends a response to a request-based event from an external application to the Orchestration Platform. SCXML State Block Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram.	Attach Block	
enables you to specify multiple application routes based on a branching condition. Depending on which condition is satisfied, the call follows the corresponding application route. Cancel Event Block Detach Block Use to cancel a custom event. Use for detaching a specific interaction from the current Orchestration Server session. Use to invoke the Cancel Call treatment, which ends the strategy and deletes the interaction from URS memory. ECMAScript Block End Parallel Block Luse to build an ECMA Script expression for routing decisions. End Parallel Block Entry Block Entry Block All routing strategy diagrams must start with an Entry block. Defines variables that can be shared across different blocks in the same workflow. The Entry block cannot have any incoming connections. Exit Block Use to terminate the workflow or to return control back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. Looping Block In cases where multiple records are returned, the Looping block can loop through all the records. Raise Event Block Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram.	Begin Parallel Block	
Detach Block Use for detaching a specific interaction from the current Orchestration Server session. Use to invoke the Cancel Call treatment, which ends the strategy and deletes the interaction from URS memory. ECMAScript Block End Parallel Block Entry Block Exit Block Use to terminate the workflow or to return control back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. Looping Block In cases where multiple records are returned, the Looping block can loop through all the records. Raise Event Block Response Block Sends a response to a request-based event from an external application to the Orchestration Platform. SCXML State Block Supposition Block Use to invoke external SCXML document that it generates based on the workflow diagram. Supposition Block Use to invoke external SCXML documents or a sub-	Branching Block	enables you to specify multiple application routes based on a branching condition. Depending on which condition is satisfied, the call follows the
Disconnect Block Disconnect Block Disconnect Block Disconnect Block ECMAScript Block End Parallel Block Entry Block	Cancel Event Block	Use to cancel a custom event.
Disconnect Block ECMAScript Block End Parallel Block Entry Block Exit Blo	Detach Block	
decisions. End Parallel Block Entry Block Exit Block	Disconnect Block	ends the strategy and deletes the interaction from
Entry Block Entry Block Entry Block Entry Block Exit Block Log Block Looping Block Looping Block Response Block Response Block SCXML State Block Entry Block Entry Block Started by a matching Begin Parallel block. All routing strategy diagrams must start with an Entry block. Defines variables that can be shared across different blocks in the same workflow. The Entry block cannot have any incoming connections. Use to terminate the workflow or to return control back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. In cases where multiple records are returned, the Looping block can loop through all the records. Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Subroutine Block Use to invoke external SCXML documents or a sub-	ECMAScript Block	
Entry Block Entry block. Defines variables that can be shared across different blocks in the same workflow. The Entry block cannot have any incoming connections. Use to terminate the workflow or to return control back to calling workflow in case of subworkflow (subroutine). Log Block Log Block Looping Block Looping Block Looping Block Raise Event Block Response Block Response Block SCXML State Block SCXML State Block Entry block. Defines variables that can be shared across different blocks in the same workflow. The Entry block calling workflow or to return control back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. In cases where multiple records are returned, the Looping block can loop through all the records. Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Subroutine Block Use to invoke external SCXML documents or a sub-	End Parallel Block	
back to calling workflow in case of subworkflow (subroutine). Use to log information about the application; for example, caller-recorded input that is collected while the application is running, or error messages. Looping Block Looping Block In cases where multiple records are returned, the Looping block can loop through all the records. Raise Event Block Use to throw custom events. Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Subrouting Block Use to invoke external SCXML documents or a sub-	Entry Block	Entry block. Defines variables that can be shared across different blocks in the same workflow. The
Looping Block Looping Block Looping Block Looping Block Raise Event Block Response Block SCXML State Block Looping Block Looping Block Looping block can loop through all the records. Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Use to invoke external SCXML documents or a sub-	Exit Block	back to calling workflow in case of subworkflow
Looping Block Raise Event Block Use to throw custom events. Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Subroutine Block Looping block can loop through all the records. Use to throw custom events. Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram.	Log Block	example, caller-recorded input that is collected
Response Block Sends a response to a request-based event from an external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Subroutine Block Use to invoke external SCXML documents or a sub-	Looping Block	
external application to the Orchestration Platform. Use to write custom SCXML code for Composer to include in the SCXML document that it generates based on the workflow diagram. Subroutine Block Use to invoke external SCXML documents or a sub-	Raise Event Block	Use to throw custom events.
include in the SCXML document that it generates based on the workflow diagram. Subroutine Block Use to invoke external SCXML documents or a sub-	Response Block	
	SCXML State Block	include in the SCXML document that it generates
	Subroutine Block	

Composer Help 2

User Data	Use to update an interaction's User Data.
Wait Event Block	Use to have ORS transition out when one of the defined events is received and the associated condition is true.

Composer Help 3