

# **GENESYS**<sup>®</sup>

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.

# Working with the iWD Business Process in Composer

# Changes to IWDBP Strategies & Subroutines in 8.5.105

#### Important

The details in this topic concern changes made to the IWD BP for Composer/ORS in release 8.5.105. Strategies and subroutines not referenced here remain the same as in release 8.5.104.

Code has been refactored in order to simplify IWD strategies.

# Prioritization Strategy

## Prioritization Strategy

The purpose of this strategy is to invoke the corresponding prioritization rules, analyze the result of the rules application and place the interaction into the appropriate queue, depending on the result.

This strategy processes interactions from the following queues:

- iwd\_bp\_comp.Main.iWD\_Captured—Interactions have to satisfy the following conditions:
  - Active interactions only (interactions which do not have the property IWD\_activationDateTime set, or this property has a time stamp which is in the past.
  - Interactions are taken in the order they were submitted.

#### Changes in 8.5.105

Code that was previously in the Prioritization strategy has been moved to the InvokeGRE and InvokeUCS strategies.

# Composer Configuration

	۲		View	Properties		* 🗆 X.
	Configure View Op	ptions				
	Name O, ActivateInteract	Add <u>B</u> emove Remove All	Main Parameter Enabled: Name: Description: Check interval:	ized Conditions Segmental C ActivateInteractionsOnly IWD_activationDateTime 0	tion	asn't set up at i
Condition settings		+ = x	Condition:	IWD_activationDateTime	is null OR (_curren	t_tin 🗾
Value: IWD_activationDateTime is null OR (_current_tim	ne() >= IWD_activation	DateTime)	Order: Scheduling: Database hints:	IWD_activationDateTime a	asc	
					Cancel	ОК
	Cancel	ОК				

# Flow Summary



- 1. Entry to Prioritization workflow.
- 2. The InvokeGRE subroutine is invoked.
- 3. Log message in case if interaction was from some reasons deleted.
- 4. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_GRE\_Error

- vInLastErrorString—Error description that occurred in InvokeGRE subroutine.
- 5. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_ErrorHeld queue.
- 6. Exit Prioritization workflow.

## Distribution Strategy

## Distribution Strategy

This strategy routes interactions to a requested Agent, requested Agent Group, requested Skill, or to the default iWD Agent Group. This strategy processes interactions from the following queues:

- iwd\_bp\_comp.Main.iWD\_Queued—Interactions have to satisfy the following conditions:
- Interactions that are not subject for immediate reprioritization (interactions that do not have the property IWD\_reprioritizeDateTime set, or that have this property set to a time stamp that is in the future).
- Interactions are taken in order of priority (highest priority first)

## Changes in 8.5.105

• A Segmentation feature has been added to the Distribution routing strategy in the iWD Business Processes for Composer/ORS. Segmenting interactions ensures that all agents are kept busy by distributing tasks in each segment separately. As a result, even in a Distribution strategy that is populated by high-priority tasks assigned to small groups of agents, the strategy will not become so saturated that distribution of tasks to other agents is blocked.

Segmentation settings have been added to the **ToDistribute** view of the Distribution routing strategy. The Distribution strategy can now make a call to the segmentation setting and add an IWD\_Segment attribute to the interaction data.

# Composer Configuration - Segmentation View

🖲 View Properties Opl-byd-moli2						
Configure View Options						
			(Commentation)			
Name		Configured Segments				
ToReprioritize [iwd_bp_comp =	Bernove	Name	Value	Add		
	Remove All	MD_ext_requestedAgentG	iroup IND_ext_requestedAge	IntGroup Edd		
				Bemove		
				Remove All		
		4				
		Segment interval: 0		(seconds)		
		Segment limit: 200				
		Default Segment limit: 100				
· · ·						
(?)				Cancel OK		

Flow Summary

#### Part 1

Click to enlarge.





Click to enlarge.



#### Part 3

Click to enlarge.



- 1. Entry to Distribution workflow.
- 2. A variables are initialized:
  - vRequestedAgentGroup—Read from task attribute IWD\_ext\_requestedAgentGroup
  - vRequestedAgentGroup—Read from task attribute IWD\_ext\_requestedAgent
  - vRequestedSkill—Read from task attribute IWD\_ext\_requestedSkill
  - vCurrentTint—Current time in seconds
  - vReprioritizeDint—Read from task attribute IWD\_businessValue
  - vDefaultTargetTimeout—Default target timeout set to 3600 seconds
  - vInxPriority—Read from task attribute Priority
- 3. Delete IWD\_Route\_Error from attached data. Calculate WaitTarget timeout based on vReprioritizeDTInt and vCurrentDTInt. Sets URS priority.
- 4. Set information about clear IWD\_Route\_Error attribute.
- 5. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_Error
  - vInLastErrorString—Error description: 'Update IWD\_Route\_Error timeout'

- 6. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_ErrorHeld queue.
- Calculate vWaitTargetTimeout.
- 8. Check if calculated vWaitTargetTimeout is in range (0, vDefaultTargetTimeout>.
- 9. Set vWaitTargetTimeout to vDefaultTargetTimeout.
- 10. Exit Distribution workflow.
- 11. Check if particular Agent is requested.
- 12. Assign vRequestedAgent + '.a' to vRequestedAgent variable.
- 13. Set vIWDSegment to '\_requested\_agent'.
- 14. Route interaction to requested vRequestedAgent without waiting.
- 15. Set vIWDSegment to '\_requested\_skill'.
- 16. Route interaction to requested vRequestedAgent with requested skill without waiting.
- 17. Check if particular AgentGroup is requested.
- 18. Assign vRequestedAgentGroup + '.qa' to vRequestedAgentGroup variable.
- 19. Set vIWDSegment to '\_requested\_agent\_group'.
- 20. Route interaction to requested vRequestedAgentGroup with vWaitTargetTimeout.
- 21. Set vIWDSegment to 'default'.
- 22. Route interaction to IWD Agent Group with vWaitTargetTimeout.
- 23. Log message in case if interaction was from some reasons deleted.
- 24. Assign last route interaction error to vLastError.
- 25. Exit Distribution workflow.
- 26. Check if route interaction finished with an error.
- 27. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_Route\_Error
  - vInLastErrorString—Error description that occurred in route interaction
- 28. Exit Distribution workflow.

#### Invoke GRE Strategy

#### Invoke GRE Strategy

#### Important

**For Composer/ORS versions prior 8.1.400.48**—If custom task attributes will be used in the Standard Rules Template, you must add them in the External Service block called InvokeGRE in the InvokeGRE workflow. All user-defined attributes need to be added in the User Data attribute, otherwise they will not be attached to the task and so will not be sent in the ESP request to the external ESP service.

#### Changes in 8.5.105

Code that was previously in the Prioritization strategy has been moved to the InvokeGRE and InvokeUCS strategies.

## Composer Configuration

ł	🛞 User Data @pl-byd-mo02							
Configure User Data								
	Set the User Data to be passed to the specified external service.							
-								
	Кеу	Value		<u>A</u> dd				
	MediaType	∨ariable(∨MediaType)		Edit				
	IWD_reprioritizeDateTime	Variable(vIwdReprioritizeDateTime)		<u></u> uit				
	IWD_activationDateTime	Variable(vIwdActivationDateTime)		<u>R</u> emove				
	IWD_businessCalendarId	∨ariable(vIwdBusinessCalendarId)		Bernove All				
	IWD_businessValue	∨ariable(vIwdBusinessValue)						
	IWD_capturePointId	∨ariable(vIwdCapturePointId)		⊻ariables				
	IWD_category	∨ariable(vIwdCategory)						
	IWD_channel	∨ariable(vIwdChannel)						
	IWD_dueDateTime	Variable(vIwdDueDateTime)						
	Priority	Variable(vPriority)						
	ReceivedAt	Variable(vReceivedAt)						
				<u>D</u> own				
	(?)		Cancel	ок				

# Flow Summary

#### Part 1



Part 3



- 1. Entry to InvokeGRE strategy.
- 2. Check if in\_method\_name is set to SetBusinessContext or Prioritize.
- 3. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_GRE\_Error
  - vInLastErrorString—Error informs that: vInMethodName + 'is not valid'
- 4. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_ErrorHeld queue.
- 5. Exit InvokeGRE workflow.
- 6. The **FindListObjectItem** subroutine is invoked to determine the name of the Genesys Rules Engine Application. The subroutine uses the List Object list **GREServerList**:
  - vInItemName—GREServerList
  - vInListName—Iwd\_Esp\_List
- Check if vInCustomPackageName was published to this subroutine. If it is set then vInCustomPackageName will be run. Otherwise package name needs to be found in Iwd\_Package\_List.
- 8. Assign vInCustomPackageName to vGrePackageName.
- 9. Delete IWD\_GRE\_Result, IWD\_Error, RulePhase before Invoke GRE.
- 10. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_GRE\_Determination\_Error
  - vInLastErrorString—Error description that occurred in FindListObjectItem subroutine.
- 11. The FindListObjectItem subroutine is invoked to determine the name of the rule package that the Genesys Rules Engine will be invoking to evaluate the classification rules:

- vInItemName—RulePackageList
- vInListName—Iwd\_Package\_List
- 12. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_Rule\_Package\_Determination\_Error
  - vInLastErrorString—Error description that occurred in FindListObjectItem subroutine.
- 13. An ESP request is sent to the Genesys Rules Engine to evaluate the classification rules.

#### Important

All user data that needs to be added to ESP request must be added in User Data attributes.

- 14. Parse ESP result and attach to the interaction all attributes modified by the GRE.
- 15. Invoke **AssignLastError** subroutine with attributes:
  - vInLastErrorkey—IWD\_GRE\_Error
  - vInLastErrorString—Error informs that: 'Attach GreResult timeout'
- 16. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_ErrorHeld queue.
- 17. Exit InvokeGRE workflow.
- 18. CheckBusinessValueAndPriority subroutine is called to verify if IWD\_businessValue and Priority have correct values.
- 19. Check if in\_method\_name is set to SetBusinessContext or Prioritize.
- 20. Check if IWD\_processId was set by any rules or when task was created.
- 21. Check is made to see if this is the first time that prioritization rules are being evaluated for the interaction, and the priority was not set up by any rules.
- 22. Get last error that was occured in GRE call and assign it to vLastError variable.
- 23. A check is done to see if the error code is related to the ESP server communication.
- 24. A delay is introduced, based on the value of the \_delay\_ms variable. The flow goes back to step 11 to retry the connection to the ESP server.
- 25. The last Interaction Server-related error is extracted from a variable.
- 26. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_GRE\_Error
  - vInLastErrorString—The last Interaction Server-related error is extracted from a variable.
- 27. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_GRE\_Error
  - vInLastErrorString—The last Interaction Server-related error is extracted from a variable.
- 28. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_GRE\_Error

- vInLastErrorString—The last Interaction Server-related error is extracted from a variable
- 29. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_Rejected queue.
- 30. Exit InvokeGRE workflow.
- 31. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_Prioritization\_Error
  - vInLastErrorString—Error description: 'Priority is not set up by rules'.
- 32. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_Queued queue.
- 33. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_Captured queue.
- 34. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_ErrorHeld queue.
- 35. Exit InvokeGRE workflow.

#### CheckBusinessValueandPriority Subroutine

#### CheckBusinessValueAndPriority Subroutine

The purpose of this workflow is to verify if Priority and IWD\_businessValue have correct values.

#### Changes in 8.5.105

Code has been refactored in order to simplify this IWD strategy.

# Flow Summary

#### Part 1







- 1. Entry to CheckBusinessValueAndPriority workflow.
- 2. Variables are initialized:
  - vIwdBusinessValue—Read from task attribute IWD\_businessValue
  - vIwdPriority—Read from task attribute Priority
- 3. Validate if vIwdBusinessValue is valid.
- 4. Set vIwdBusinessValue to vMinBusinessValue.
- 5. Set vIwdBusinessValue to vMaxBusinessValue.

- 6. Update IWD\_businessValue to vIwdBusinessValue.
- 7. Validate if vIwdPriority is valid.
- 8. Set vIwdPriority to vMinPriority.
- 9. Set vIwdPriority to vMaxPriority.
- 10. Update Priority to vIwdPriority.
- 11. Exit CheckBusinessValueAndPriority workflow.
- 12. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_Error
  - vInLastErrorString Error description: 'Update Priority timeout'
- 13. Invoke AssignLastError subroutine with attributes:
  - vInLastErrorkey—IWD\_Error
  - vInLastErrorString— Error description: 'Update iWD\_businessValue timeout'
- 14. The interaction is placed in the iwd\_bp\_comp.Main.iWD\_ErrorHeld queue.
- 15. Exit CheckBusinessValueAndPriority workflow.