



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

IWDBP Strategies & Subroutines in 8.5.104

Classification Strategy

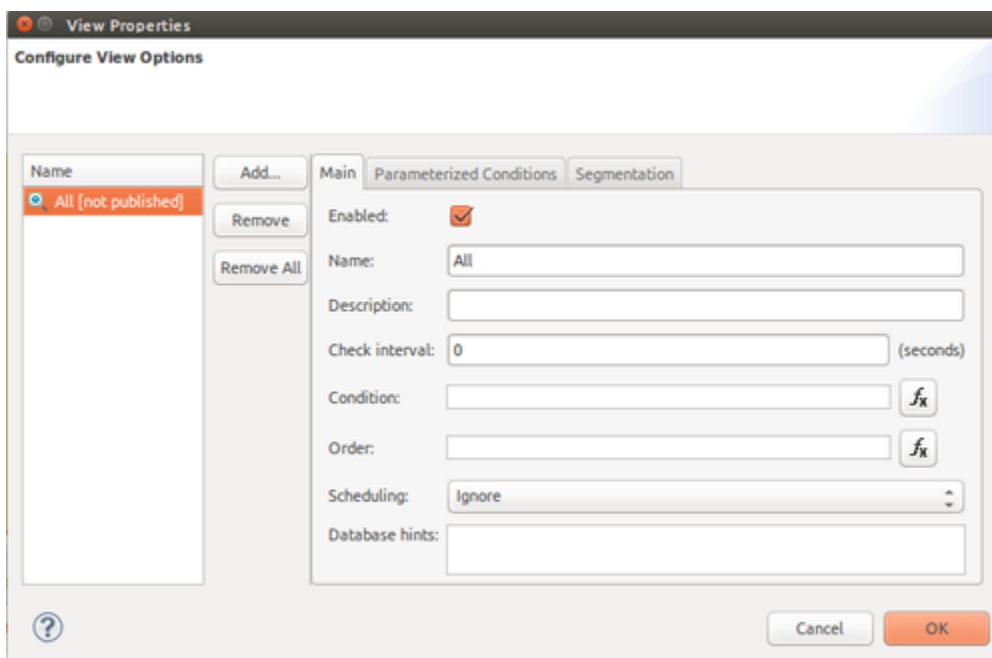
Classification Strategy

The purpose of this strategy is to invoke corresponding classification 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_New`—Interactions have to satisfy the following conditions:
 - There are no conditions here.
 - Interactions are taken in order they were submitted.

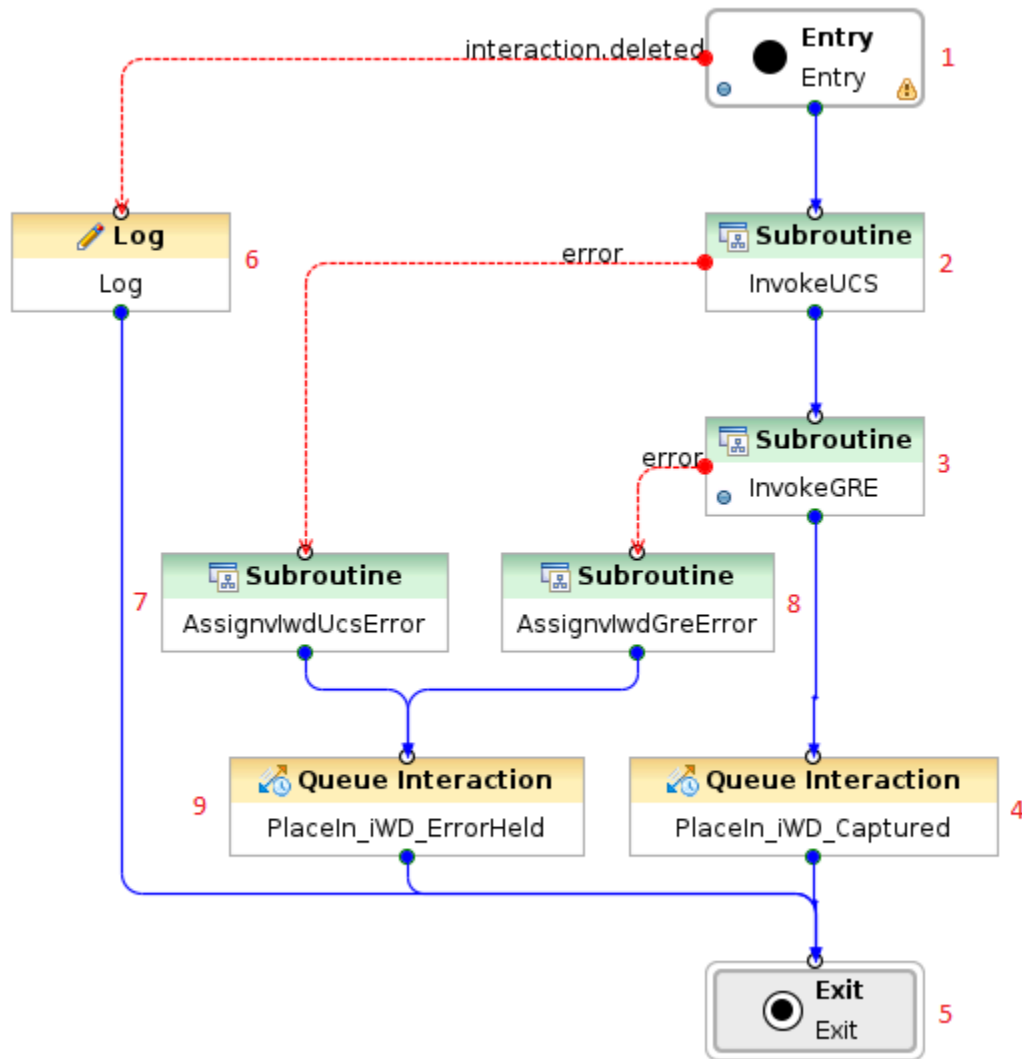
Composer Configuration



The Composer configuration for this strategy block shows that all interactions are distributed to the

iwd_bp_comp.Main.iWD_New queue without conditions.

Flow Summary



Flow Detail

1. Entry to Classification workflow.

2. The InvokeUCS subroutine is invoked to create new interaction in the UCS database.
3. The InvokeGRE subroutine is invoked.
4. The interaction is placed in the `iwd_bp_comp.Main.iWD_Captured` queue.
5. Exit Classification workflow.
6. Log message in case if interaction was from some reasons deleted.
7. Invoke AssignLastError subroutine with attributes:
 - `vInLastErrorkey—IWD_UCS_Error`.
 - `vInLastErrorString`—Error description that occurred in InvokeUCS subroutine.
8. Invoke AssignLastError subroutine with attributes:
 - `vInLastErrorkey—IWD_GRE_Error`
 - `vInLastErrorString`—Error description that occurred in InvokeGRE subroutine.
9. The interaction is placed in the `iwd_bp_comp.Main.iWD_ErrorHeld` queue.

Prioritization Strategy

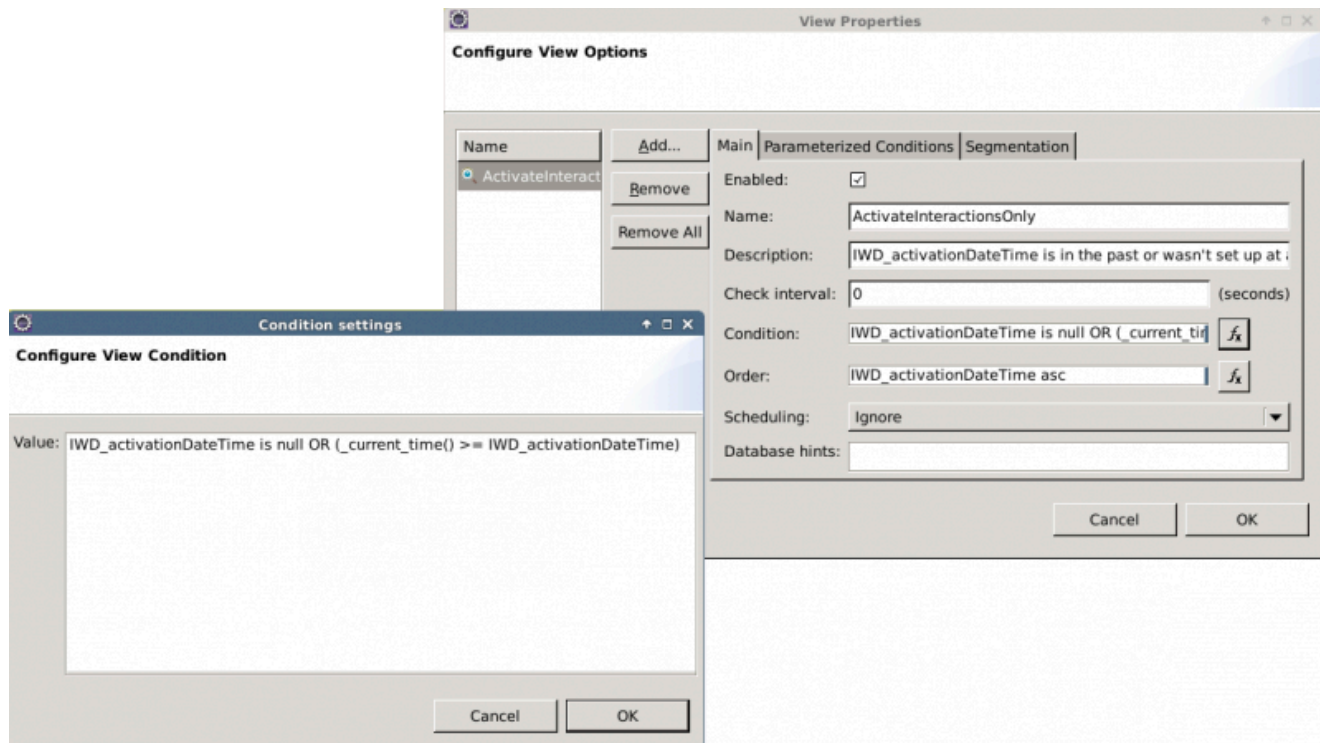
Prioritization Strategy

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.

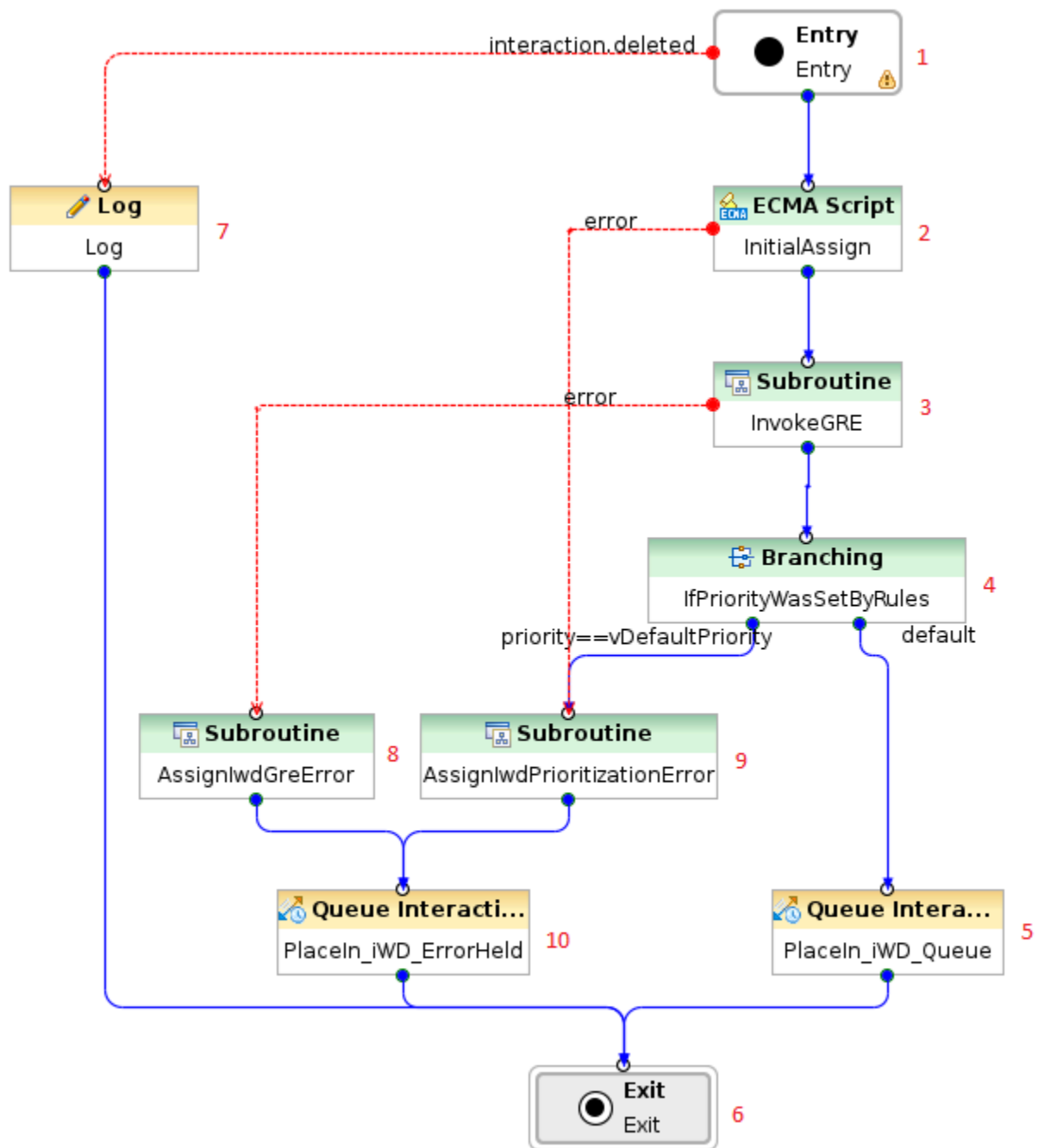
This strategy processes interactions from the following queues:

- `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.

Composer Configuration



Flow Summary



Flow Detail

1. Entry to Prioritization workflow.
2. A variable is initialized:
 - `vSourceQueue`—Read from task attribute.
3. The `InvokeGRE` subroutine is invoked.
4. 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.
5. The interaction is placed in the `ibd_bp_comp.Main.ibd_Queue` queue.
6. Exit Prioritization workflow.
7. Log message in case if interaction was from some reasons deleted.
8. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey`—`IBD_GRE_Error`
 - `vInLastErrorString`—Error description that occurred in `InvokeGRE` subroutine.
9. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey`—`IBD_Prioritization_Error`
 - `vInLastErrorString`—Error description that occurred in initialization or prioritization rules were not evaluated.
10. The interaction is placed in the `ibd_bp_comp.Main.ibd_ErrorHeld` queue.

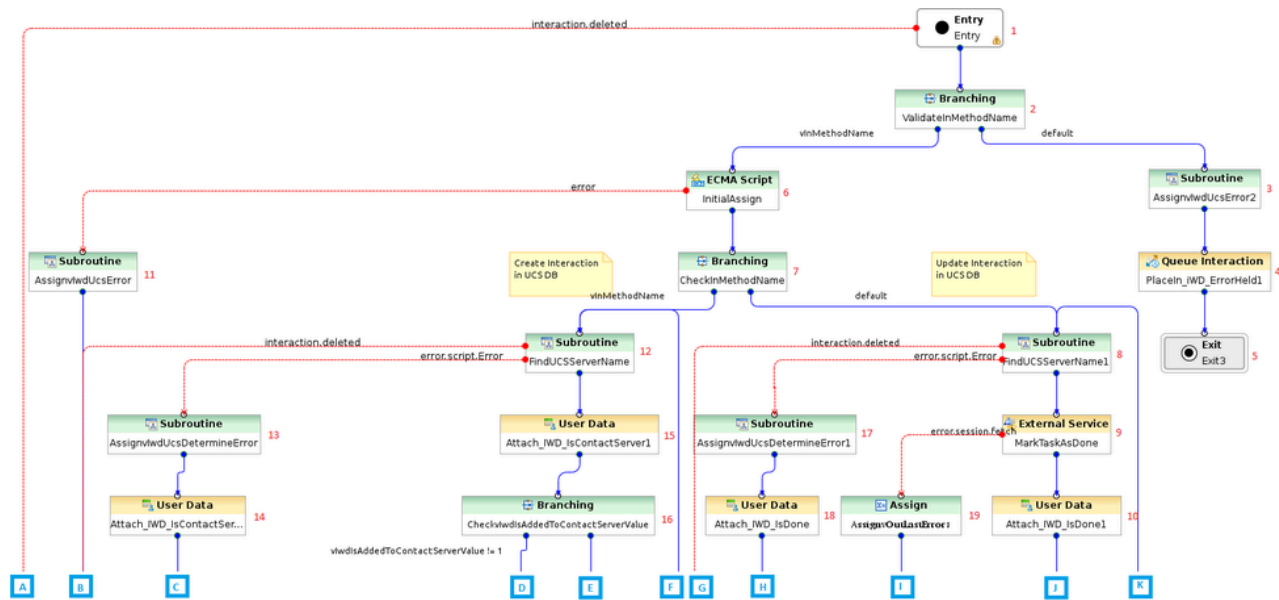
Invoke UCS Strategy

Invoke UCS Strategy

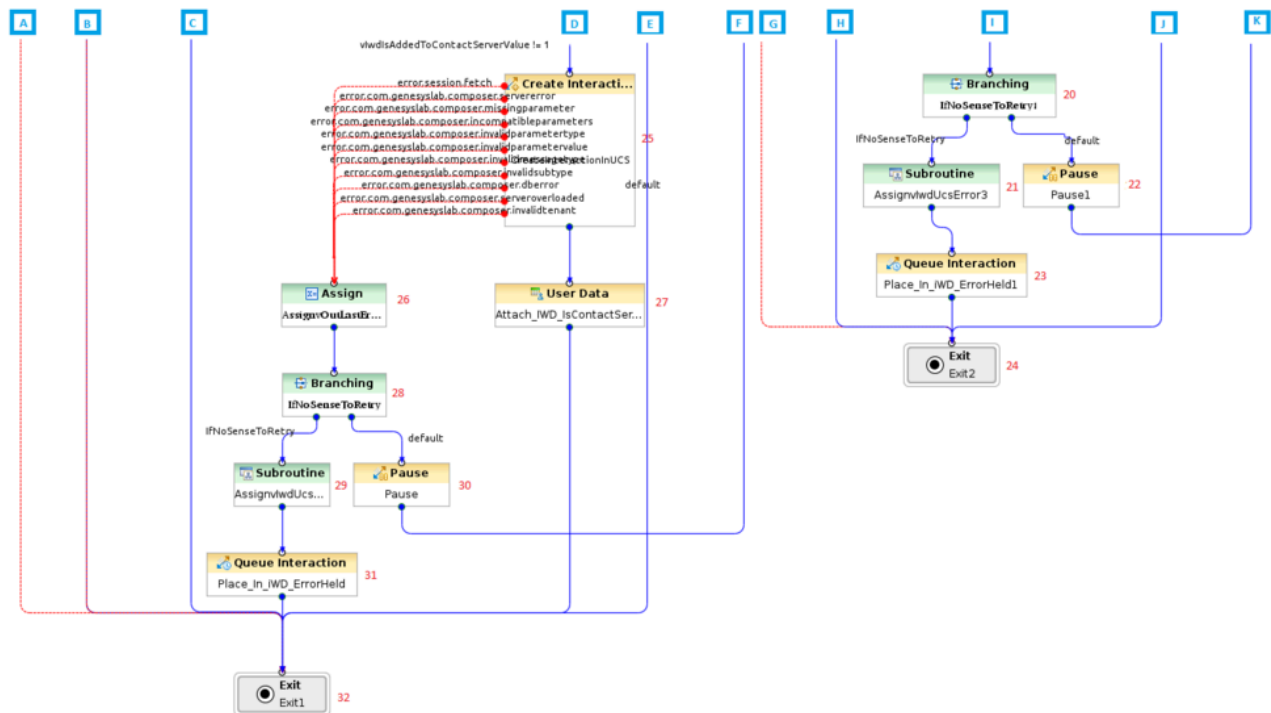
The purpose of this workflow is to create an interaction in the UCS database if UCS is configured.

Flow Summary

Part 1



Part 2



Flow Detail

1. Entry InvokeUCS strategy.
 2. Check if `in_method_name = 'Create' | in_method_name = 'OMInteractions'`.
 3. Invoke AssignLastError subroutine with attributes:
 - `vInLastErrorkey—IWD_UCS_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 InvokeUCS workflow.
 6. A variables are initialized:
 - `vExternalId—Read from task attribute ExternalId`
 - `vMediaType—Read from task attribute`
 - `vSubmittedBy—Read from task attribute attr_itx_submitted_by`
 - `vType—Read from task attribute 'InteractionType'`
 - `vSubType—Read from task attribute 'InteractionSubtype'`
 - `vIwdIsAddedToContactServerValue—Read from task attribute 'IWD_isAddedToContactServer'`
 7. Check if `in_method_name = 'Create'`.
 8. The FindListItem subroutine is invoked to determine the name of the UCS Application. The subroutine uses the List Object list GREServerList:
 - `vInItemName—ContactServerList`
 - `vInListName—Iwd_Esp_List`
 9. The strategy calls a method on the Universal Contact Server to set the status of the interaction to 3, indicating that the interaction is done.
 10. The value of the user data key `IWD_isDone` is set to 1.
 11. Invoke AssignLastError subroutine with attributes:
 - `vInLastErrorkey—IWD_UCS_Error'`
 - `vInLastErrorString—Error description that occurred when variables were initialized`
 12. The FindListItem subroutine is invoked to determine the name of the UCS Application. The subroutine uses the List Object list GREServerList:
 - `vInItemName—ContactServerList`
 - `vInListName—Iwd_Esp_List`
 13. Invoke AssignLastError subroutine with attributes:
 - `vInLastErrorkey—IWD_UCS_Determination_Error'`
 - `vInLastErrorString—Error description that occurred in FindListItem subroutine.`
 14. The value of the user data key `IWD_isContactServer` is set to 0.
-

15. The value of the user data key `IWD_isContactServer` is set to 1.
16. Check if `IWD_isContactServer` is set to 1.
17. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey—IWD_UCS_Determination_Error'`
 - `vInLastErrorString—`Error description that occurred in `FindListObjectItem` subroutine.
18. The value of the user data key `IWD_isDone` is set to 0.
19. An error is extracted from user data and assigned in `vLastError` variable.
20. If it makes sense to retry updating the interaction record in UCS.
21. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey—IWD_UCS_Error'`
 - `vInLastErrorString—`Information that it does not make sense to retry update interaction in UCS.
22. A delay is introduced into the processing. Flow returns to step 8.
23. The interaction is placed in the `ibd_bp_comp.Main.iWD_ErrorHeld` queue.
24. Exit `InvokeUCS` workflow.
25. A new interaction is created in the UCS database, for this iWD task.
26. An error is extracted from user data and assigned in `vLastError` variable.
27. The user data key `IWD_isAddedToContactServer` is updated to 1 to indicate that the task was successfully added to the interaction history in UCS. The result returned from the ESP call to UCS (from See A new interaction is created in the UCS database, for this iWD task. If that function is successful is written to the variable `IWD_UCS_Result`.
28. If it makes sense to retry creating the interaction record in UCS.
29. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey—IWD_UCS_Error`
 - `vInLastErrorString—`Information that it does not make sense to retry create interaction in UCS
30. A delay is introduced into the processing. Flow returns to step 12.
31. The interaction is placed in the `ibd_bp_comp.Main.iWD_ErrorHeld` queue.
32. Exit `InvokeUCS` workflow.

Invoke GRE Strategy

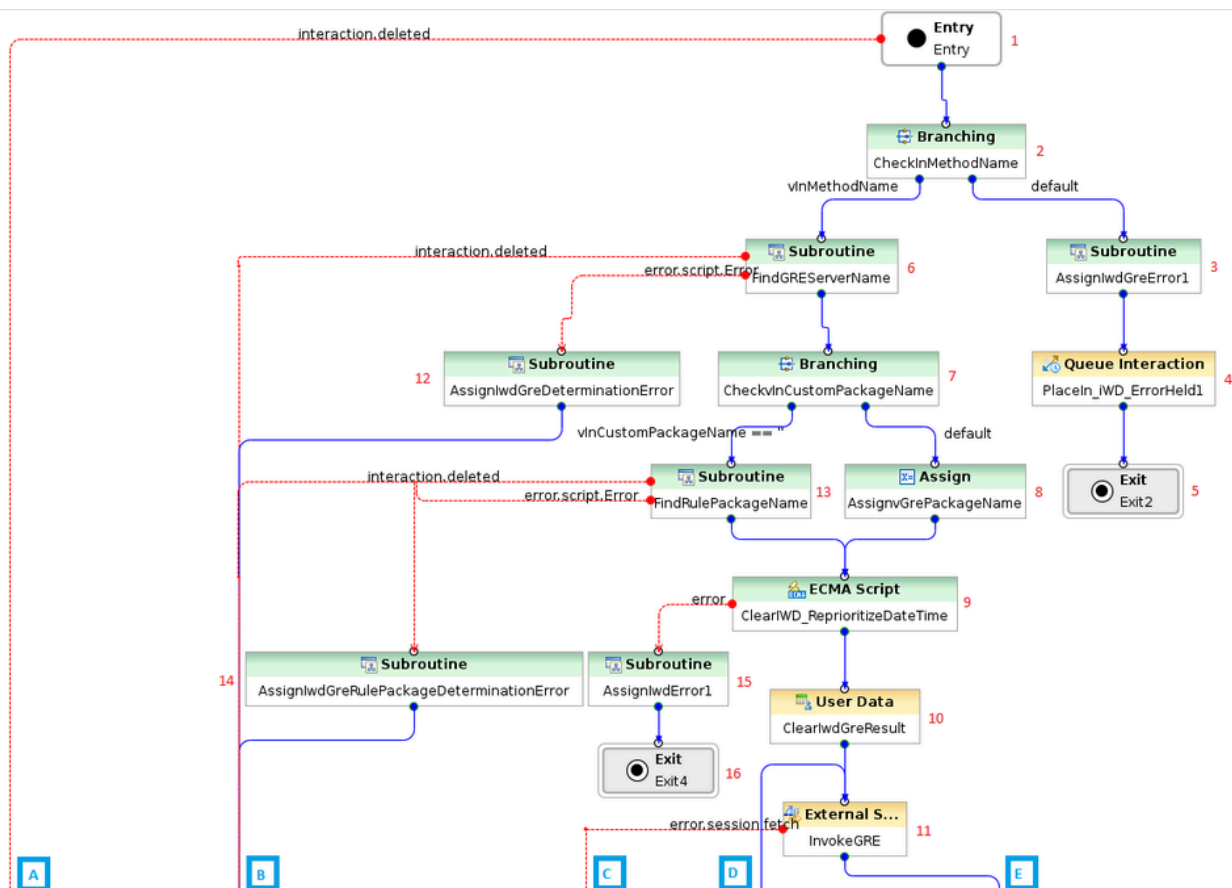
Invoke GRE Strategy

Important

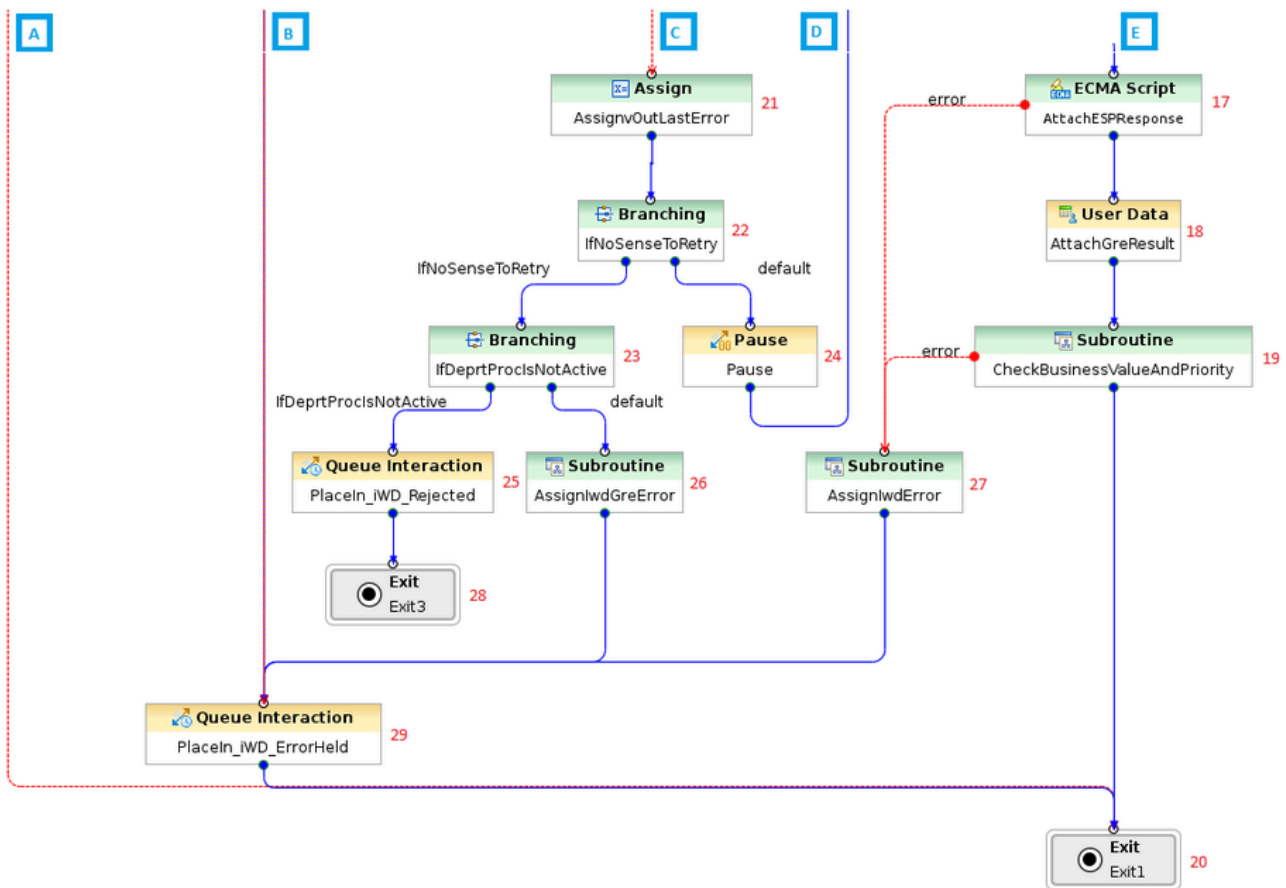
For Composer/ORS versions prior 8.1.400.48—If custom task attributes will be used in the Standard Rules Template, you must add them in the External Service block called InvokeGRE in the InvokeGRE workflow. All user-defined attributes need to be added in the User Data attribute, otherwise they will not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

Flow Summary

Part 1



Part 2



Flow Detail

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 FindListItem 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`
7. 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_reprioritizeDateTime` from attached data.
 10. Delete `IWD_GRE_Result` before Invoke GRE.
 11. An ESP request is sent to the Genesys Rules Engine to evaluate the classification rules.
 12. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey—IWD_GRE_Determination_Error`
 - `vInLastErrorString`—Error description that occurred in `FindListObjectItem` subroutine
 13. 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`
 14. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey—IWD_Rule_Package_Determination_Error`
 - `vInLastErrorString`—Error description that occurred in `FindListObjectItem` subroutine
 15. Invoke `AssignLastError` subroutine with attributes:
 - `vInLastErrorkey—IWD_Error`
 - `vInLastErrorString`—Error description that occurred when `IWD_reprioritizeDateTime` was cleared
 16. Exit `InvokeGRE` workflow.
 17. Parse ESP result and assign it to `vGreResult` variable.
 18. The ESP result is attached to user data as a key-value pair with the key `IWD_GRE_Result`.
 19. `CheckBusinessValueAndPriority` subroutine is called to verify if `IWD_businessValue` and `Priority` have correct values.
 20. Exit `InvokeGRE` workflow.
 21. Get last error that was occurred in GRE call and assign it to `vLastError` variable.
 22. A check is done to see if the error code is related to the ESP server communication.
 23. The last Interaction Server-related error is extracted from a variable.
 24. A delay is introduced, based on the value of the `_delay_ms` variable. The flow goes back to 11 to retry the connection to the ESP server.
 25. The interaction is placed in the `iwd_bp_comp.Main.iWD_Rejected` queue.
 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_Error
 - vInLastErrorString—Error description that occurred when we parsed ESP result
28. Exit InvokeGRE workflow.
29. The interaction is placed in the `iwd_bp_comp.Main.iWD_ErrorHeld` queue.

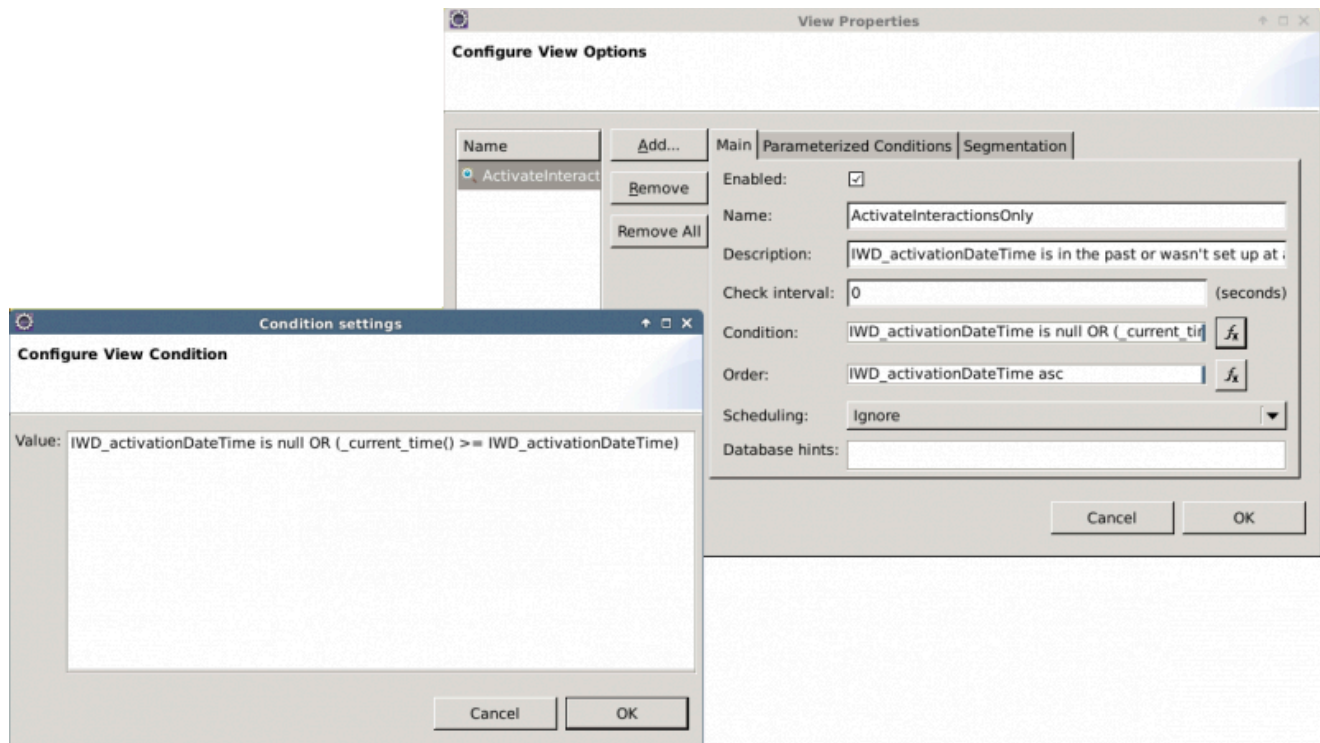
Distribution Strategy

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:

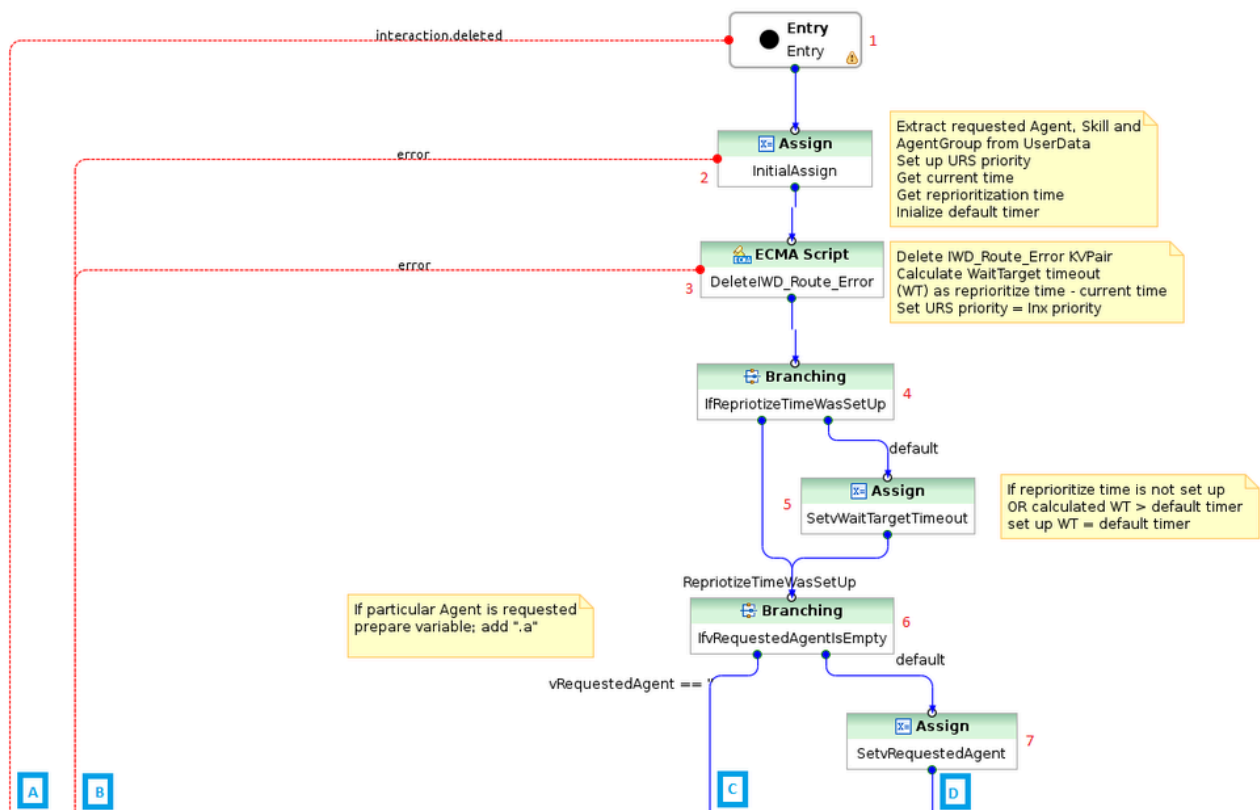
- `iwd_bp_comp.Main.iWD_Queued`—Interactions have to satisfy the following conditions:
- Interactions that are not subject for immediate reprioritization (interactions that do not have the property `IWD_reprioritizeDateTime` set, or that have this property set to a time stamp that is in the future).
- Interactions are taken in order of priority (highest priority first)

Composer Configuration

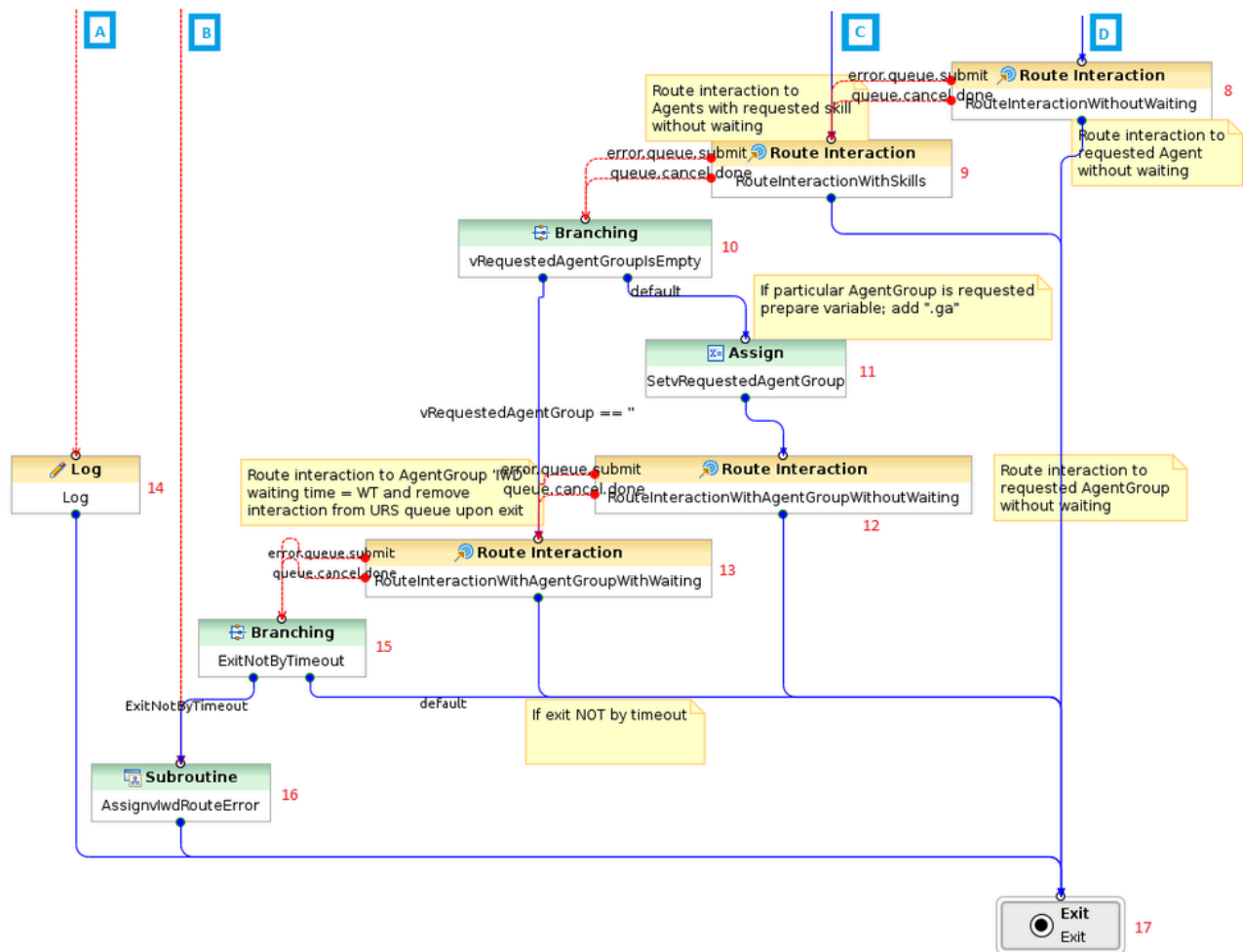


Flow Summary

Part 1



Part 2



Flow Detail

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
 - vReprioritizedDint—Read from task attribute IWD_businessValue
 - vDefaultTargetTimeout—Default target timeout set to 3600 seconds

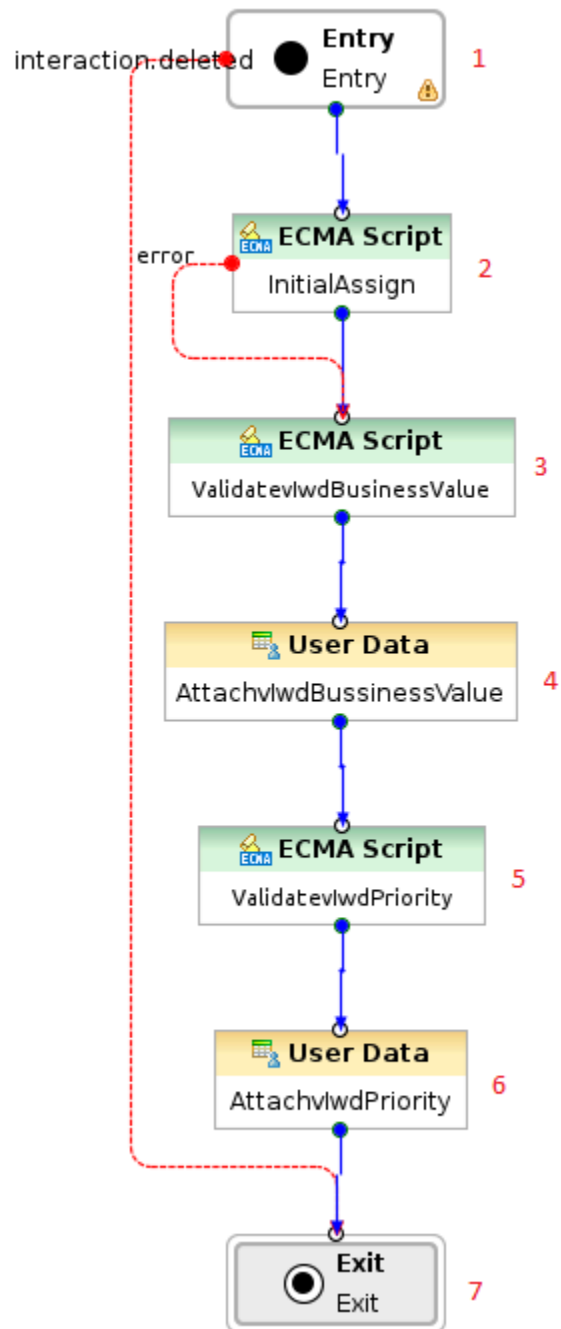
- vInxPriority—Read from task attribute Priority
3. Delete IWD_Route_Error from attached data. Calculate WaitTarget timeout based on vReprioritizedDTInt and vCurrentDTInt. Sets URS priority.
 4. Check if reprioritize time was set up and calculated vWaitTargetTimeout <= vDefaultTargetTimeout.
 5. If reprioritize time is not set or vWaitTargetTimeout > vDefaultTargetTimeout then set vWaitTargetTimeout to vDefaultTargetTimeout.
 6. Check if particular Agent is requested.
 7. Assign vRequestedAgent + '.a' to vRequestedAgent variable.
 8. Route interaction to requested vRequestedAgent without waiting.
 9. Route interaction to requested vRequestedAgent with requested skill without waiting.
 10. Check if particular AgentGroup is requested.
 11. Assign vRequestedAgentGroup + '.qa' to vRequestedAgentGroup variable.
 12. Route interaction to requested vRequestedAgentGroup without waiting.
 13. Route interaction to requested vRequestedAgentGroup with waiting.
 14. Log message in case if interaction was from some reasons deleted.
 15. Check if route interaction finished with an error.
 16. Invoke AssignLastError subroutine with attributes:
 - vInLastErrorkey—IWD_Route_Error
 - vInLastErrorString - Error description that occurred in route interaction
 17. Exit Distribution workflow.

CheckBusinessValueAndPriority Subroutine

CheckBusinessValueAndPriority Subroutine

The purpose of this workflow is to verify if Priority and IWD_businessValue have correct values.

Flow Summary



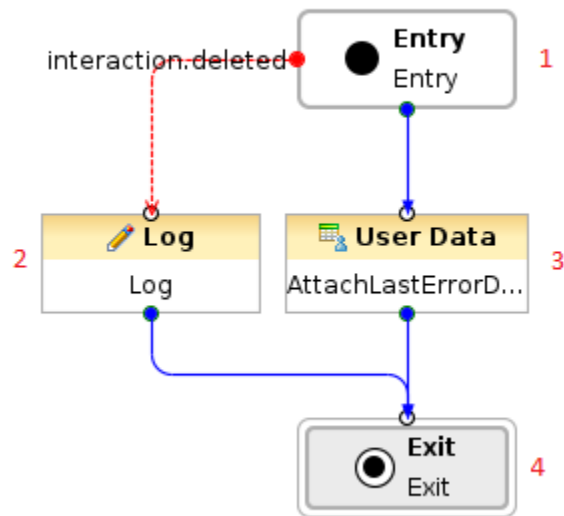
Flow Detail

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. The vIwdBusinessValue is attached to user data with the key IWD_businessValue and value vIwdBusinessValue.
5. Validate if vIwdPriority is valid.
6. The vIwdPriority is attached to user data with the key Priority and value vIwdPriority.
7. Exit CheckBusinessValueAndPriority workflow.

AssignLastError Subroutine

AssignLastError Subroutine

Flow Summary



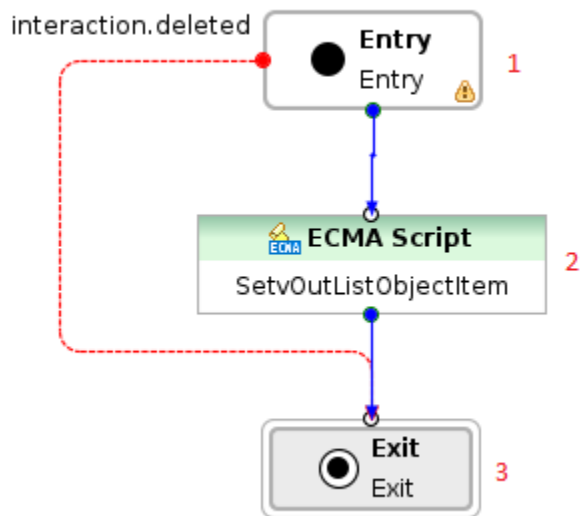
Flow Detail

1. Entry to AssignLastError workflow.
2. The last error is attached to user data as a key-value pair with the key `vInLastErrorkey` and value `vInLastErrorString`.
`vInLastErrorkey` and `vInLastErrorString` are workflow attributes that need to be set before calling this workflow.
3. Log message if interaction was from some reasons deleted.
4. Exit AssignLastError workflow.

FindListItem Subroutine

FindListObjectItem Subroutine

Flow Summary



Flow Detail

1. Entry to FindListObjectItem workflow.
2. Search vKeyToFindInListObject in vInListName.
 - vInItemName—Section in vInListName
 - vInListName—List object where vKeyToFindInListObject should be searched
 - vKeyToFindInListObject—Option in vInItemName that should be found
3. When vKeyToFindInListObject is found in vInListName, then the value assigned to this option will be assigned to vOutListObjectItem.
4. Exit FindListObjectItem workflow

MarkInteractionAsDone Strategy

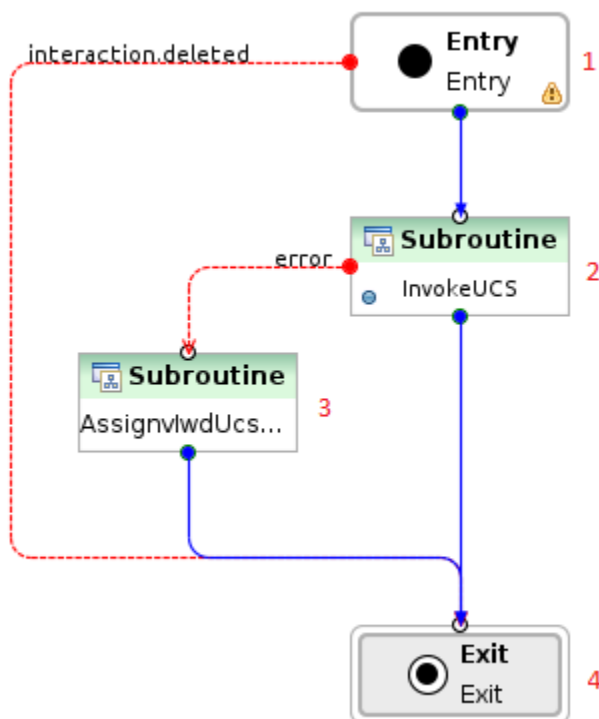
MarkInteractionAsDone Strategy

The purpose of this strategy is to update the Universal Contact Server (UCS) database to mark the interaction as done. This equates to setting the value in the Status column of the Interactions table to 3. UCS clients, such as Interaction Workspace, will then display the status of this interaction as done when the user looks at interactions they have previously processed.

Interactions have to satisfy the following conditions:

- The value of the attached data key IWD_isContactServer is 1
- The value of the attached data key IWD_isDone is either null or 0 (zero)

Flow Summary



Flow Detail

1. Entry to MarkInteractionAsDone workflow.
 2. The InvokeUCS subroutine is invoked to complete interaction in the UCS database.
-

3. Invoke AssignLastError subroutine with attributes:
 - vInLastErrorkey—IWD_UCS_Error
 - vInLastErrorString—Error description that occurred in InvokeUCS subroutine
4. Exit MarkInteractionAsDone workflow.

Removal Strategy

Removal Strategy

The purpose of this strategy is to delete expired interactions from the Interaction Server database.

A key-value pair in user data with the key `IWD_expirationDateTime` contains information about when an interaction has to be deleted.

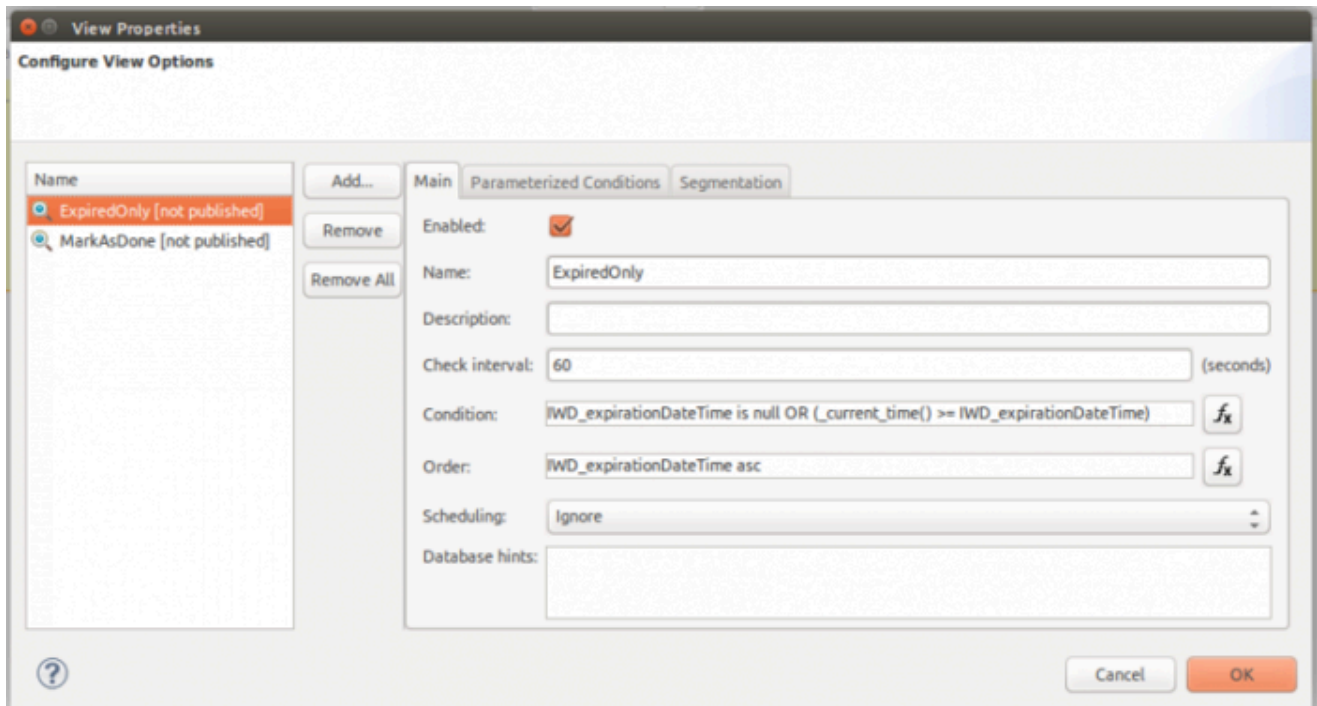
This strategy processes interactions from the following queues:

- `iwd_bp_comp.Main.iWD_Completed`
- `iwd_bp_comp.Main.iWD_Canceled`
- `iwd_bp_comp.Main.iWD_Rejected`

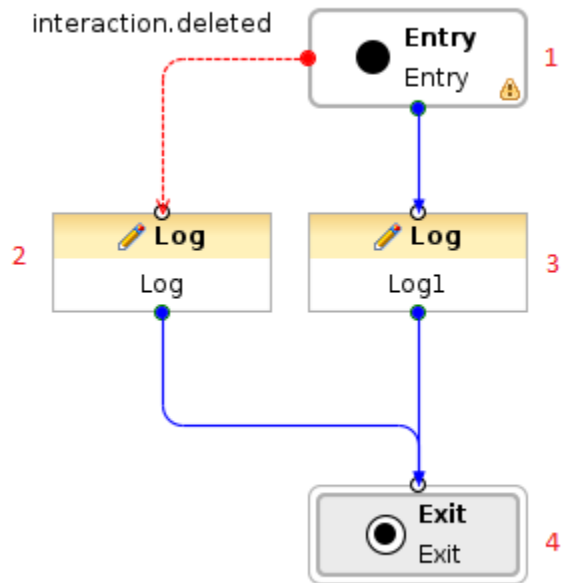
Interactions have to satisfy the following conditions:

- Interactions must either have the property `IWD_expirationDateTime` not set, or this property must have a time stamp which is in the past.
- Interactions are taken in the order they were submitted.

Composer Configuration



Flow Summary



Flow Detail

1. Entry to Removal workflow.
2. Log message in case if interaction was from some reasons deleted.
3. Log message: Task will be terminated on exit
4. Exit Removal workflow,

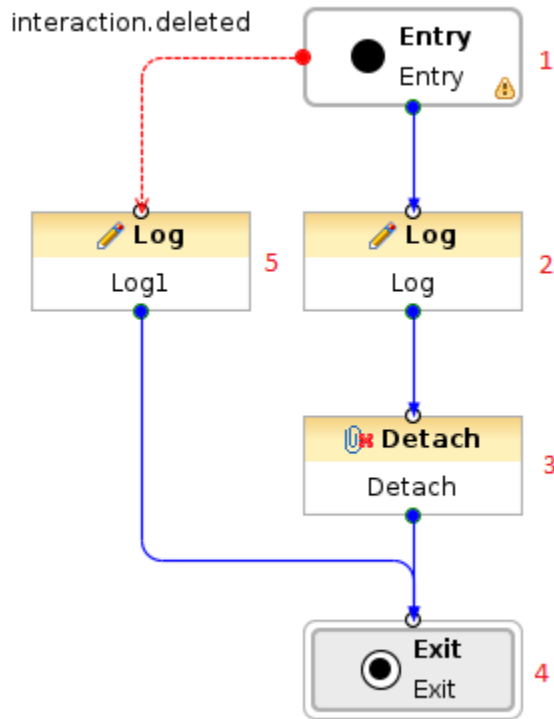
Finish Strategy

Finish Strategy

This workflow detaches interactions from the session. This workflow processes interactions from the following queues:

- `iwd_bp_comp.Main.iWD_Errorheld`

Flow Summary



Flow Detail

1. Entry to Finish workflow.
2. Log message : 'Task processing completed'.
3. Detach interaction. After this operation interaction will not be processed any more.
4. Log message in case if interaction was for some reason deleted.
5. Exit Finish workflow.