Composer Help

Publishing Updates

5/11/2025

Publishing Updates

Contents

Publishing an interaction process diagram validates Project configuration information and pushes the information out to Configuration Server. When you configure an Interaction Queue, Workflow, or Workbin block, Composer does not send the information to Configuration Server until you invoke the Publish operation. This gives you complete control of the update process. When publishing an interaction process diagram, Composer also updates the Configuration Server/Object Name property of the IPD blocks for which a configuration object was created (applies to Workflow, Workbin, Interaction Queue blocks). You can set Configuration Server Preferences to:

- Automatically publish upon saving.
- Display a prompt to save before publishing.
- Delete published objects when an interaction process diagram is deleted.
- Delete published objects when a project is closed or deleted.

For information on resetting IPD Publish information, see the figure in topic Project Properties dialog box. Expand `Reset IPD Publish Information`.

When to Publish

- Any time properties for any of the blocks in the IPD are changed or blocks are added/removed.
- For example, to create a Submitter Script object in the Configuration Database.
- Any time a workflow diagram is renamed. In such cases, you must go back to the Workflow block in the IPD diagram and point the block to the renamed workflow.
- If you rename your workflow Project -- this will change the deployment URL for the Project. Publishing again will point the enhanced routing object to the new URL.
- If you delete Published objects in Configuration Server, you can re-publish the diagram to create new objects.

Note: If routing blocks refer to previously unpublished queues, these references may become incorrect when the queue is published. These errors are caught by workflow validations and should be fixed by selecting the published queue in any block properties that show this validation error.

How to Publish

1. Before publishing, you must connect to Configuration Server.
2. Right-click an interaction process diagram in the Project Explorer.
3. Select Publish to Configuration Server. The following message appears: The selected IPD diagram's data was successfully published to Configuration Server.

Once these objects are created successfully in Configuration Server, some manual configuration is still required before interactions can work. For example, to redirect e-mails to an Endpoint, you must set endpoint key in the pop-clientX section of the e-mail server Application to the correct end-point. For more details, see Deploying a Routing Application.

Composer Marker Section

The Publish operation puts a marker section in each Configuration Database object it creates. The section name is __COMPOSER__. The section has the following keys:

Key Name	Value	Description
source	composer	Hard-coded value to indicate object was created by Composer
last_updated	<timestamp>	Indicates the timestamp when the publish operation was initiated that modified this object. Example: Thu Jan 07 11:26:36 PST 2010. I
owner_diagram	<string>	Name of the Composer diagram that published this object.
owner_project	<string>	Name of the Composer project that owns the IPD diagram in owner_diagram.
owner_uuid	<string>	Identifier of the Composer diagram that published this object.

IRD Objects Not Modified

The Publish operation does not modify queue and view objects that were not created by Composer. This avoids accidentally modifying objects previously created by Universal Routing's Interaction Routing Designer (IRD), which may cause issues in a legacy IRD/URS system. Using the marker section described above to indicate where the object was created, the following types of objects are not modified if they were not created by Composer:

- CfgScript of type Interaction Queue
- CfgScript of type Interaction Queue View
- CfgScript of type Workflow

Media Server Block Update Detail

The following updates are written to Configuration Server for all Media Server blocks in all IPD diagrams in the Project:

- Updates endpoints:<tenant DBID> section of specified CfgApplication.
- Each endpoint name attribute's value is set to the name of the interaction queue's CfgScript object that it is connected to. Only endpoints that were owned by this diagram, or were previously unconnected, are updated.

Interaction Queue Block Update Detail

The following updates are written to Configuration Server for all [blocks in all IPD diagrams in the Project:

- A CfgScript object (type = Interaction Queue) is created under the current tenant/Scripts folder. If the object already exists, its properties are updated.

Block Property	Configuration Server Object/Option
Description	CfgScript object/Annex Section Queue/Description property
Queue Enabled	State property of the CfgScript object. If TRUE, set to CFGEnabled. If FALSE, set to CFGDisabled.

- In the Annex of the CfgScript object, the property application is created in section Orchestration. Composer writes the value in this format: script:<name of Enhanced Routing CfgScript object where <name of EnhancedRouting CfgScript object> will be replaced with the workflow name.

This will essentially cause all views of the interaction queue to submit interactions to the SCXML application that the Enhanced Routing object points to. In the IPD, this will be a Workflow block pointing to an existing workflow diagram or an SCXML file. Note: If you rename an Interaction Queue block after its corresponding CfgScript object has been created, the object name in Configuration Server remains unchanged. Instead, the key Name in the Annex section Namespace and its value are set to the new name. Composer displays the changed name.

Interaction Queue View Property Update Detail

The following updates are written to Configuration Server for all Views defined for all Interaction Queue blocks and Workbin blocks in all IPD diagrams in the Project:

- A `CfgScript` object (type = Interaction Queue View) is created under the current tenant/`Scripts` folder. If the object already exists, its properties are updated. The name of the object follows this format: <container queue CfgScript object name>/<view name>

Block Property	Config. Server Object Type	Annex/Option Section	Key/Property
View Name	CfgScript	Namespace	Name
Description	CfgScript	View	Description
Enabled	CfgScript	-	State property
Check Interval	CfgScript	View	Freeze Interval
Condition	CfgScript	View	Condition
Order	CfgScript	View	Order

Scheduling	CfgScript	View	scheduling-mode
Parameterized Conditions (multi-valued)	CfgScript	View	Each value creates a key like "Condition.<value>"
Database Hints (Oracle)	CfgScript	View	sql-hint
Segmentation (multi-valued)	CfgScript	View	segment-by Value will be "value1,value2,...valueN"
Segment Check Interval	CfgScript	View	segment-check-interval
Segment Limit	CfgScript	View	segment-total-limit

Note: If you rename a View after its corresponding CfgScript object has been created, the object name in Configuration Server remains unchanged. Instead, the key Name in the Annex section Namespace and its value are set to the new name. Composer displays the changed name.

Workflow Block Update Detail

The following updates are written to Configuration Server for all Workflow blocks in the Project.

- A `CfgScript` object of the Enhanced Routing type is created under the current tenant/Scripts folder.
- In its annex, property url is created in section Application. The value is the URL of the generated SCXML document on the Composer web server (bundled Tomcat or local IIS). You can change this property using Configuration Manager or Genesys Administrator to set the correct value for the deployment environment.

In its annex, in section `ApplicationParams`, the key CustomerView_URL is added. Its value is in this format: [http:// http://]<configured CV host>:<configured CV port>. Context Services port and host values are picked up from Composer preferences.

Workflow Blocks and Publishing an IPD

This use cases below apply when Use Interaction Submitters is not enabled and you are not using Interaction Submitters.

In this case, when publishing an interaction process diagram (IPD) to Configuration Server, Workflow blocks are handled in two different ways:

Use Case #1: The Workflow block is dedicated to voice or interaction-less processing. In that case, you must use a stand-alone block (the block is not connected to any other block).

- When generating the code: Composer generates one SCXML file per such Workflow block (Name= IPD_<ipd file name>_<workflow block name>.scxml).
- When publishing: Composer creates one Enhanced Routing Script object (Name=<Project Name>.<IPD name>.<Workflow block name>) per such Workflow block in the IPD being published. The Application/url property of the ERS refers to the SCXML url (if deployed). The Workflow block Object Name property (read only property) is updated to the name of the Enhanced Routing Script object.

Use Case #2: The Workflow block is dedicated to multimedia processing. In this case, the block is connected (directly or indirectly) after a Workbin block or an Interaction Queue block.

- When generating the code: Composer generates one SCXML file per Interaction Queue/Workbin block (Name=IPD_<ipd file name>_<interaction queue block name>.scxml). If an IPD has an Interaction Queue block connected to multiple Workflow blocks (multiple views are defined on the Interaction Queue block), only one SCXML file is generated when generating the code for that IPD. This unique IPD SCXML is used to initiate the execution for all Workflow blocks. At runtime, the Workflow SCXML to execute is selected depending on the view the interaction is pulled from.

- When publishing: Composer creates one Interaction Queue Script object (Name=<Project Name>.<IPD name>.<Interaction Queue block name>) per Interaction Queue block. Composer creates one Interaction Queue View Script object (Name=<Project Name>.<IPD name>.<Interaction Queue block name>.<View name>) per Interaction Queue block defined view.

Composer creates one Enhanced Routing Script object (Name=<Project Name>.<IPD name>.<Interaction Queue block name>.Routing) per Interaction Queue block in the IPD being published. The Application/url property of this Enhanced Routing Script object refers to the Queue IPD SCXML url (if deployed). Composer does NOT create an Enhanced Routing Script object for the workflow blocks. The Workflow block Object Name property (read only property) is NOT updated.

Workbin Block Update Detail

The following updates are written to Configuration Server for all Workbin blocks in the Project.

- A `CfgScript` object of the Interaction Workbin type is created under the current tenant/Scripts folder. If the object already exists, its properties are updated.
- A `CfgScript` object of the Interaction Queue type is created under the current tenant/Scripts folder. The name of the object follows this format: <Workbin CfgScript object name>.PrivateQueue.
- A `CfgScript` object of the Interaction Queue View type is created under the current tenant/Scripts folder. The name of the object follows this format: <Workbin CfgScript object name>.PrivateView.
- A `CfgScript object of the Enhanced Routing type is created under the current tenant/Scripts folder. The name of the object follows this` format: <Workbin CfgScript object name>.PrivateQueue.Routing.
- A `CfgScript` object of the Interaction Queue View type is created under the current tenant/Scripts folder for each of this workbin user defined view. The name of the object follows this format: <Workbin CfgScript object name>.<view name>.

Starting with Composer 8.1.410.14, Composer generates one Submitter for each not-private view (<project name>.<IPD name>.<Workbin name>.<View name>.submitter). Each Submitter is published with the following parameters:

```
Submitter>Strategy=<project name>.<IPD name>.<Workbin name>.PrivateQueue.Routing
Submitter>View=<project name>.<IPD name>.<Workbin name>.<View name>
```

Deleting Items from Configuration Server

Configuration Server objects (Enhanced Routing, Interaction Queue, Interaction Queue View, Workbin) might be deleted from the Configuration Server in following situations:

- An interaction process diagram is deleted and that IPD contained blocks for which configuration objects were created. See also Configuration Server Preferences, Delete published objects, when Interaction Process Diagram is closed or deleted option. Note that Composer must be connected to the Configuration Server in order to be able to effectively delete the Configuration objects.
- A project containing interaction process diagrams is closed or deleted and those IPD contained block(s) for which configuration objects were created. See also see Configuration Server Preferences, Delete published objects when Project is closed or deleted option. Note that Composer must be connected to the Configuration Server in order to be able to effectively delete the Configuration objects.
- An interaction process diagram is published and
  - some blocks of that IPD, for which configuration objects were created, have been deleted.
  - some blocks of that IPD, for which configuration objects were created, have been updated (for example some views were removed from an Interaction Queue block).

In all cases, the user is prompted for deletion confirmation for each Configuration Server object that Composer is going to delete.