

# OpenText Communications Orchestration

Gain insights into communications generated to ensure compliance and transparency

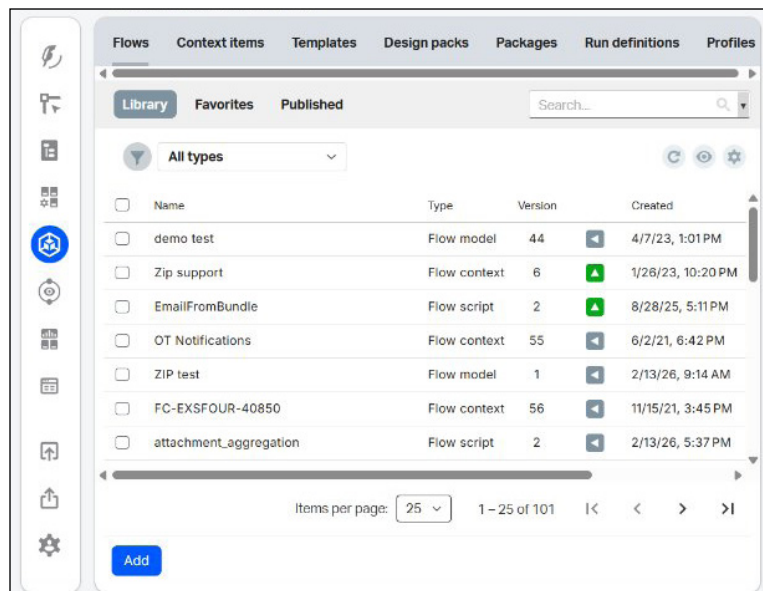
## Benefits

- Design communications workflows in a single thin-client interface
- Gain operational insights with dashboards
- Ensure successful delivery of every communication

Organizations face growing challenges in managing high-volume, multichannel customer communications due to fragmented systems with limited visibility into communication workflows. These issues often result in inconsistent messaging, delayed delivery, increased operational costs, and difficulty meeting regulatory requirements. OpenText™ Communications, with its built-in orchestration capabilities, addresses these challenges by providing a centralized platform to design, automate, and monitor communication flows. It enables organizations to automate document generation, manage communication channels, and orchestrate end-to-end messaging workflows.

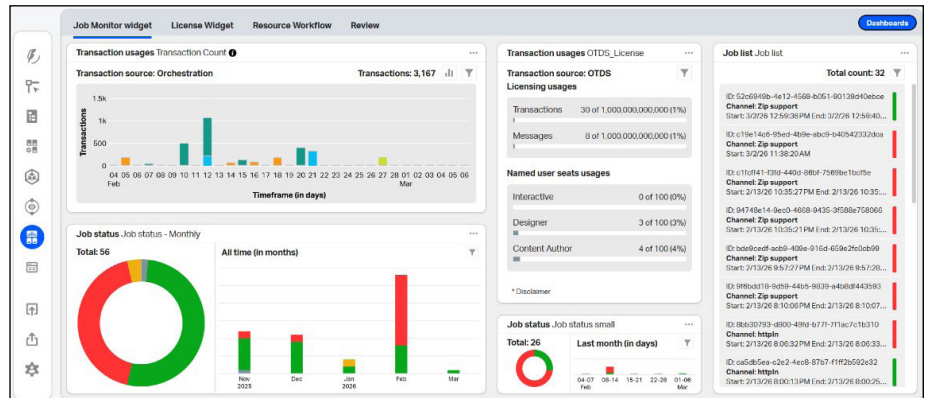
## Design communications workflows in a single thin-client interface

Design, manage, and execute end-to-end communication workflows through a unified thin-client interface, eliminating complexity while increasing operational efficiency. Process owners can design visual communication flows that support end-to-end communication creation, from data intake through final delivery using an intuitive flow model. The orchestration asset library further enhances flexibility by providing the flow context, where users can configure connectors; define process automation functions, such as sorting and bundling; and implement customized scripts through Flow Logic to address unique production processing.



## Gain operational insights with dashboards

Ever-increasing communication generation demands make it harder for operations managers to monitor what is happening and when. Accelerate data-driven decisions with an operational dashboard that lets you monitor job performance, view transaction usage, manage resource workflows, and review outputs before delivery.

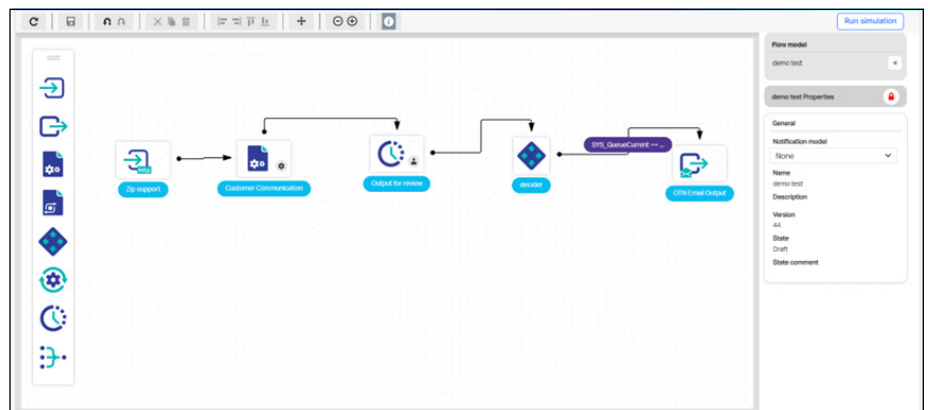


## Ensure successful delivery of every communication

Ensure critical communications reach their intended recipients with assured message delivery, through OpenText Communication integration with OpenText™ Core Messaging. Reliable digital delivery maximizes success through automated fallback mechanisms, seamlessly redirecting failed email messages to an alternate channel, such as SMS.

## Flow model

- Flow models define the communication workflow. Each model typically begins with an input channel that receives data (e.g., via HTTPS POST).
- Data is routed to a communication node where the OpenText Communications engine generates required documents.
- Output is delivered via output channel nodes (e.g., email, API response, repositories).



Flow model

## Alert and notifications models

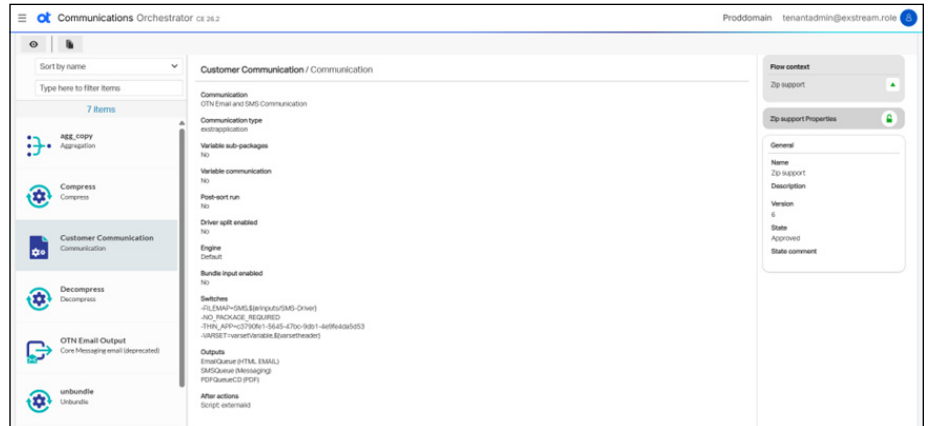
- Alert and notification models are variations of flow models.
- They are triggered automatically based on events produced during communication flows.

## Flow script

- Flow scripts implement orchestration logic within a flow model.
- Example: Assigning a variable to define an external ID for a job.

## Flow context

- A flow context contains configuration items such as input channels, output channels, communications, etc.



Flow item configuration

## Input channels

Input channels enable the system to receive data from a variety of sources. The orchestrator supports the following input channel types:

<b>Azure blob storage channel</b>	Downloads files from Azure <sup>®</sup> blob storage containers.
<b>Azure service bus channel</b>	Consumes messages from Azure service bus message queues.
<b>Core Journey channel</b>	Generates additional communications based on events from OpenText <sup>™</sup> Core Journey.
<b>Core Messaging report channel</b>	Polls and processes report files generated by OpenText Core Messaging.
<b>Directory channel</b>	Monitors and scans directories for input files.
<b>Dynamic Asset Service (DAS) channel</b>	Accesses Dynamic Asset Service resources.
<b>External event queue channel</b>	Consumes events from external event queues.
<b>GCP CS channel</b>	Downloads files from Google Cloud <sup>™</sup> Storage buckets.
<b>HTTP(S) channel</b>	Invoked via HTTP POST requests.
<b>HTTP(S) download channel</b>	Accesses and downloads resources from repositories via HTTP protocol.
<b>Interactive fulfillment channel</b>	Fulfills Interactive document requests.

## Input channels

<b>Job temporary directory channel</b>	Reads content from files in the temporary root folder within the current job directory.
<b>Kafka channel</b>	Pulls messages from topics within Kafka clusters.
<b>S3 channel</b>	Downloads files from Amazon S3 buckets.
<b>SFTP channel</b>	Downloads files from directories on SFTP servers.
<b>Webhook channel</b>	Accepts POST requests to webhook URLs.
<b>Custom channels</b>	Additional channels implemented using the SDK may also be available.

## Output channels

Output channels deliver generated content from OpenText Communications to various destinations. The orchestrator supports the following output channel types:

<b>Azure blob storage channel</b>	Uploads files to Azure blob storage containers.
<b>Content Management channel</b>	Archives documents and metadata in OpenText™ Content Management.
<b>Core Archive channel</b>	Archives documents generated by Document Presentment solutions in OpenText™ Core Archive or OpenText™ Archive Center.
<b>Core Content Management channel</b>	Uploads documents and metadata to OpenText™ Core Content Management.
<b>Core Messaging email channel</b>	Delivers output to OpenText Core Messaging for email distribution to end customers.
<b>Core Messaging push channel</b>	Delivers output to OpenText Core Messaging for push notifications to end customers.
<b>Core Messaging SMS channel</b>	Delivers output to OpenText Core Messaging for SMS distribution to end customers.
<b>Core Signature channel</b>	Uploads documents with optional attachments to OpenText™ Core Signature and creates signature requests.
<b>Document response channel</b>	Returns responses to requests sent via HTTP(S) input channels.
<b>Documentum Content Management channel</b>	Transfers documents to the OpenText™ Documentum™ Content Management platform.
<b>File channel</b>	Delivers output files to directories within the base output directory for the tenant and domain.

## Output channels

<b>GCP CS channel</b>	Uploads files to Google Cloud Storage buckets.
<b>HTTP(S) channel</b>	Submits output to external systems with HTTP(S) connectivity.
<b>Information Archive channel</b>	Archives documents and metadata in OpenText™ Information Archive.
<b>Interactive editing channel</b>	Uploads documents to the Interactive repository.
<b>Job temporary files channel</b>	Writes output to temporary files within the current job folder's temporary root folder.
<b>Kafka channel</b>	Pushes messages to topics within Kafka clusters.
<b>Output Server channel</b>	Transfers documents to OpenText™ Output Server for further processing.
<b>S3 channel</b>	Uploads files to Amazon S3 buckets.
<b>SAP Document Presentment channel</b>	Delivers documents and metadata to SAP® Document Presentment; can also add OpenText Core Archive document links when used after a Core Archive channel.
<b>SFTP channel</b>	Uploads files to directories on SFTP servers.
<b>SMTP email channel</b>	Delivers output to SMTP mail servers for email distribution to end customers.
<b>WhatsApp channel</b>	Delivers messages to WhatsApp recipients.
<b>Custom channels</b>	Additional channels implemented using the SDK may also be available.

## Processors

Processors execute specific tasks within the workflow. They can be integrated as standalone flow model nodes or configured as pre/post-action triggers for other flow context items.

<b>Compress processor</b>	Compresses output following communication generation.
<b>Core Archive delete document processor</b>	Removes unlinked documents from OpenText Core Archive.
<b>Decompress processor</b>	Decompresses compressed files.
<b>Decrypt processor</b>	Decrypts encrypted data.
<b>Download Core Signature document processor</b>	Downloads signed documents and signing logs from OpenText Core Signature.

## Processors

<b>Driver split processor</b>	Determines how to split multi-customer input into batches before communication processing.
<b>Encrypt processor</b>	Encrypts data for secure transmission.
<b>Input selector processor</b>	Configured as a post-action on input channels to select the driver file from alternative locations.
<b>Report builder processor</b>	Generates reports based on variables accessible to the processor.
<b>Script processor</b>	Executes flow scripts for custom logic implementation.
<b>Unbundle processor</b>	Splits bundled messages following aggregation into individual messages per bundle element.
<b>Verify Core Signature event processor</b>	Validates that incoming data represents a legitimate OpenText Core Signature event.
<b>XSLT processor</b>	Transforms XML content using XML stylesheets.
<b>Custom processors</b>	Additional processors implemented using the SDK may also be available.

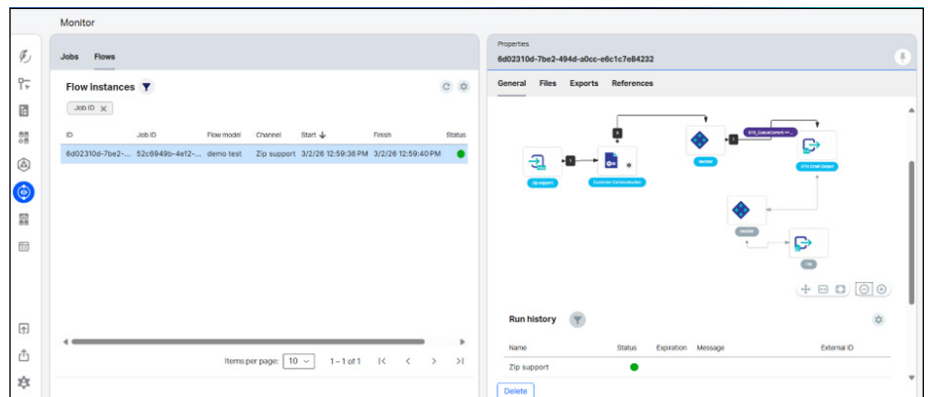
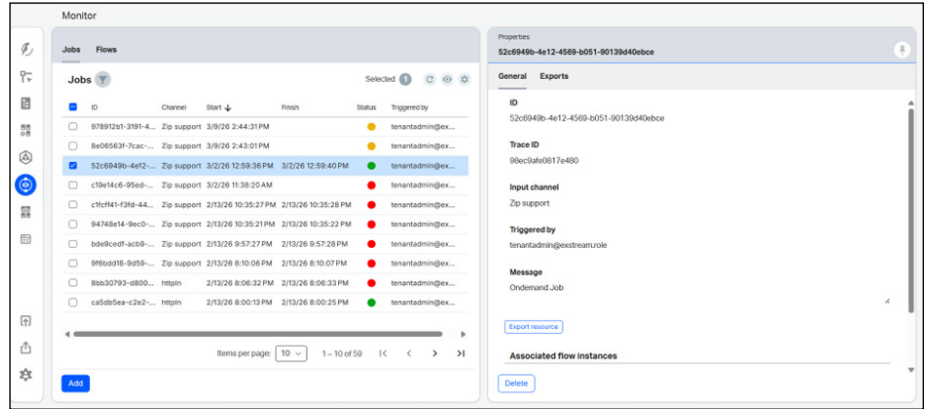
## Additional flow context Items

<b>Outcome mapping</b>	Defines response code handling for communications and output channels, specifying which response codes (success, warning, error) should trigger specific actions.
<b>Aggregation</b>	Consolidates multiple messages (documents and related assets) into bundled collections for efficient processing.
<b>Communication</b>	Integrates with the OpenText Communications engine to generate documents (communications) and custom variables.
<b>External sub-flow</b>	Connects input and output channels into specialized sub-flows. Three sub-flow types are supported: <ul style="list-style-type: none"><li>• Core Signature</li><li>• Edit Interactive document</li><li>• Sorting and bundling</li></ul>

# Job monitoring

Job Monitor provides comprehensive visibility into orchestration of job execution, displaying both currently executing jobs and historical job records with detailed information.

When data is submitted to an input channel, the system initiates new orchestration jobs for all published flow models configured to use that input channel. Each flow model execution creates a corresponding flow instance, which can be tracked through the Job Monitor interface.



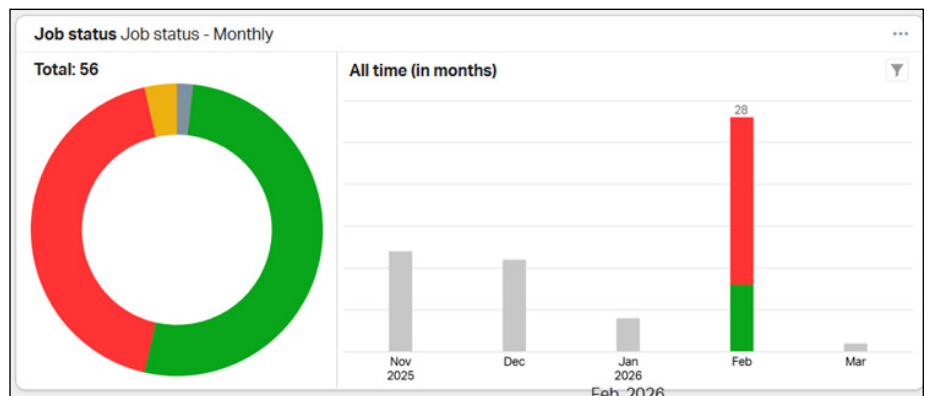
# Dashboard

The dashboard view enables the creation of multiple domain-specific dashboards populated with configurable widgets for monitoring and analysis.

The following widgets are available in dashboards view:

## Job status

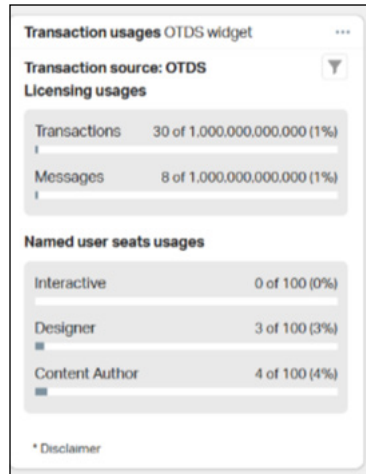
Displays a graph of the jobs run through orchestration.



Job status widget (showing data for all time)

## Transaction usage

Provides information and status about the license for the tenant from OpenText Directory Services (OTDS) or showcases transaction consumption data at domain level with filtering available on each communication level.



Transaction usage (information pulled from OTDS)



Transaction and message count widget (last month data)



## Job list

Provides information about the status of selected flow models.

ID	Channel	Start	End	Status
52c6949b-4e12-4569-b051-90139d40ebce	Zip support	3/2/26 12:59:36 PM	3/2/26 12:59:40 PM	Success
c19e14c6-95ed-4b9e-abc9-b40542332dca	Zip support	3/2/26 11:38:20 AM		Failed
c1f0ff41-f3fd-440d-86bf-7569be1bcf5e	Zip support	2/13/26 10:35:27 PM	2/13/26 10:35:28 PM	Failed
94748e14-9ec0-4668-9435-3f588e758066	Zip support	2/13/26 10:35:21 PM	2/13/26 10:35:22 PM	Failed
bde9cedf-acb9-409e-916d-659e2fc0cb99	Zip support	2/13/26 9:57:27 PM	2/13/26 9:57:28 PM	Failed
9f6bdd18-9d59-44b5-9839-a4b8df443593	Zip support	2/13/26 8:10:06 PM	2/13/26 8:10:07 PM	Failed
8bb30793-d800-49fd-b77f-7f1ac7c1b310	httpin	2/13/26 8:06:32 PM	2/13/26 8:06:33 PM	Failed
ca5db5ea-c2e2-4ec8-87b7-f1ff2b592e32	httpin	2/13/26 8:00:13 PM	2/13/26 8:00:25 PM	Success
066d81b9-f0e9-4320-adff-31b58d2e9a98	httpin	2/13/26 7:55:12 PM	2/13/26 7:55:13 PM	Success

Job list (List of job based on the filter selection)

## Resource workflow

Lists specified resource types and their workflow states. Also provides the ability to move resources through the approval workflow states.

Name	Type	Version	Updated
Claim Estimate Letter(CA)	Communication (CA)	3	8/28/25, 1:50 PM
Claims correspondence communications	Communication (CA)	1	3/7/25, 8:25 PM
PS_Pilot_LCA	Communication (CA)	2	4/13/23, 9:36 PM
Quote to Pay	Communication (CA)	2	11/13/25, 5:26 PM
AccountAdded-comSet	Run definition	1	6/4/25, 4:18 PM
BankStatement_ComSet Job	Run definition	1	2/26/21, 1:32 AM
CommSetTestMail_ComSet Job	Run definition	1	8/24/21, 6:13 PM
Core Journey	Run definition	1	9/2/24, 3:00 PM
CoreSigCommSet_ComSet Job	Run definition	1	6/8/22, 5:05 PM
CS_Requisition	Run definition	1	4/8/24, 1:41 PM
Demo Communication set_ComSet Job	Run definition	1	6/2/21, 6:29 PM
DiffTypesCommSet	Run definition	1	4/3/24, 3:27 PM
DocAddedEvent-comSet	Run definition	1	3/6/25, 6:30 PM

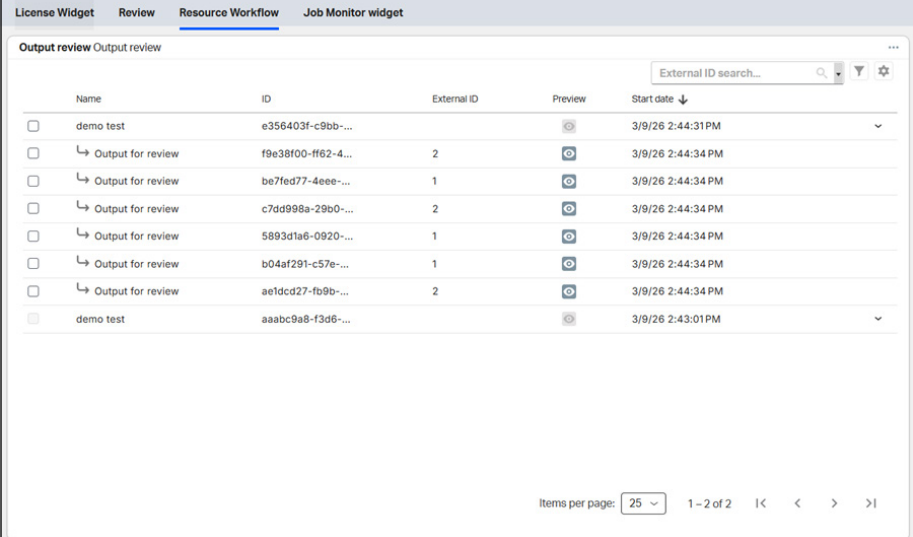
Resource workflow widget (list resource from across libraries)

## Resources

[Discover more about OpenText Communications](#) ›

## Output review

Lists all the flow steps which are on hold and provides a preview of the output.



The screenshot shows the 'Output review' widget interface. At the top, there are tabs for 'License Widget', 'Review', 'Resource Workflow' (which is selected), and 'Job Monitor widget'. Below the tabs, the title 'Output review' is displayed. A search bar labeled 'External ID search...' is located in the top right corner. The main area contains a table with the following columns: 'Name', 'ID', 'External ID', 'Preview', and 'Start date'. The table lists several items, including 'demo test' and multiple 'Output for review' entries. Each row has a checkbox on the left and a preview icon on the right. At the bottom right, there is a pagination control showing 'Items per page: 25' and '1 - 2 of 2'.

Name	ID	External ID	Preview	Start date ↓
<input type="checkbox"/> demo test	e356403f-c9bb-...			3/9/26 2:44:31 PM
<input type="checkbox"/> ↪ Output for review	f9e38f00-ff62-4...	2		3/9/26 2:44:34 PM
<input type="checkbox"/> ↪ Output for review	be7fed77-4eee-...	1		3/9/26 2:44:34 PM
<input type="checkbox"/> ↪ Output for review	c7dd998a-29b0-...	2		3/9/26 2:44:34 PM
<input type="checkbox"/> ↪ Output for review	5893d1a6-0920-...	1		3/9/26 2:44:34 PM
<input type="checkbox"/> ↪ Output for review	b04af291-c57e-...	1		3/9/26 2:44:34 PM
<input type="checkbox"/> ↪ Output for review	ae1dcd27-fb9b-...	2		3/9/26 2:44:34 PM
<input type="checkbox"/> demo test	aaabc9a8-f3d6-...			3/9/26 2:43:01 PM

Output review widget (review output before delivery)