Configure Converting Email Conversation Items to EML Files
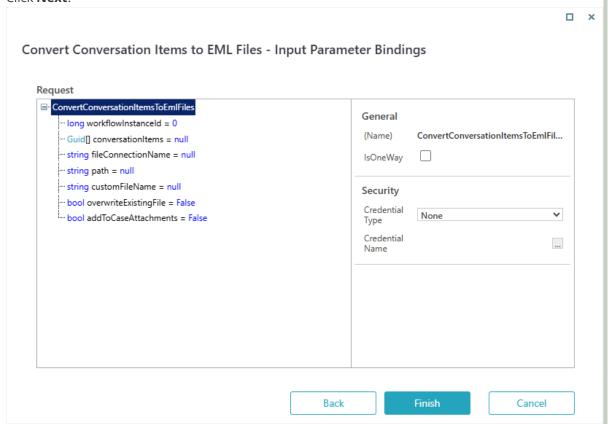
V10.9.1

Overview

For archiving and auditing purpose, as a workflow developer you can automate conversion of email conversation items into EML files at runtime by configuring a built-in command activity.

Configure converting email conversations

1. In the Administration site, open the form.
2. Add a Built-In Command activity.
3. Select **Case Operations**>**Convert Conversation Items to EML Files**.
4. Click **Next**.

5. In the Input Parameters Binding screen, add the following command parameters:
   a. workflowInstanceId: The master workflow instance Id with the Conversation View. (mandatory, value: 0).
   b. conversationItems: List of conversation item Ids. (optional, value: null).
   c. fileConnectionName: The file connection name. (optional, value: null).
      If null, then the command considers addToCaseAttachments.
   d. path: File path. (optional, value: null).
   e. customFileName: The custom file name. (mandatory, value: null).
      If null, then set the file name automatically.
   f. overwriteExistingFile: To decide if the existing file is overwritten or not. (mandatory, value: False).
      If true, then the already existing file is overwritten with no error.
   g. addToCaseAttachments: To decide if the files are added to the case attachments table of the

Conversation View master instance. (mandatory, value: False).
If true, then the files are added to the case attachments table.

6. Click **Finish**.

For details on points to consider while setting command parameters, see the key considerations section below.

Key considerations for command parameters

- If the conversationItems argument is null, then it checks for all items and prints the following.
  `Found {count} conversation items in total on Workflow instance {workflowInstanceId}`
- Then the filter is applied. If the conversation does not belong to the master workflow, then it logs the following.
  `Skipping conversation item '{itemGuid}': item does not belong to Workflow instance {workflowInstanceId}`
- If the status is not Incoming or Sent, or AutoSent is not true, then
  `Skipping conversation item '{itemGuid}': item state is {state} and AutoSent is {true/false}`
- Then it prints the count of the items it's going to convert the following.
  `Converting {count} conversation items into EML file.`
- Then, if fileConnectionName is not null, it prints the following.
  `Uploading EML files to '{fileConnectionName}'`
- If during the file upload an exception of the file storage service occurs, it will print the following and rethrow the exception:
  `Failed to upload EML files to '{fileConnectionName}'`
- Then, if it succeeded or skipped the upload to the storage, and if addToCaseAttachments is true, it prints the following.
  `Adding EML files to the attachments of workflow case '{workflowInstanceId}'`
- And, if for some reason it fails to add the attachment, it prints the following and rethrows the exception.
  `Failed to add EML files to the attachments of workflow case '{caseId}'`

Key considerations for File Name

- The following characters are invalid for file connection names, < > | : * ? \ /. These characters are converted to "_" when uploaded.
- fileConnectionName argument is not implemented as mandatory. If null value is supplied, the upload to the file storage is skipped and then files are only added to case attachments if addToCaseAttachments is true.
- This allows for the flexibility when there's no need to upload files but only convert them to EML files and store in the case.
- If invalid value of the fileConnectionName is passed, or if there's error during upload with any of the files, the exception is thrown and whole process aborts (and does not add files to the case attachments).
- The argument fileConnectionName is implemented as optional.
  - If fileConnectionName = null and addToCaseAttachments = true, then the upload to the file storage will be skipped and the files will be only added to case attachments.
  - If for the fileConnectionName an invalid value is passed, or if there's error during upload with any of the files, the system throws an exception and the whole process aborts (and does not add files to the case attachments).

Error handling

- [ ] Errors are logged to the **observability layer**.

- [ ] If any expression fails at runtime, the command execution fails.
- [ ] If the file connection or path is invalid, the command execution fails.
- [ ] If `OverwriteExistingFile = false` and the file already exists, the command execution fails.
- [ ] If converting multiple items and one fails, the entire execution fails (no partial success).