- Add the Conversations View
  With this you get all of the email functionalities. For more details, see this article.

- Manage the Email Composer controls independently through Client API within the forms.

Manage Email Composer through client API

1. Add or edit a form or task activity in a workflow.
2. Add the following script to the form source.

```
<script type="text/javascript">

  function OnNewMailButtonClientClicking(sender, args) {
    var parameters = {
      wfId: '<%= { wf.WorkflowInstanceId } %>',
      title: 'New Message',
      from: "johnk@genpact.com;",
      to: "mark.smith@genpact.digital;john.smith@pnmsoft.com",
      cc: "jack.fellow@genpact.digital",
      bcc: '',
      template: 'rat',
      subject: 'Email Composer Subject',
      body: '',
      //disabledFields: ['from', 'to', 'cc', 'bcc', 'template', 'subject']
    };
    var messageId = SEQForms.EmailComposer.open(parameters);
    args.set_cancel(true);
  }
  function OnExistingMailButtonClientClicking(sender, args) {
    var parameters = {
      wfId: '<%= { wf.WorkflowInstanceId } %>',
      messageId: <Guid>,
      template: "rat"
    }
    SEQForms.EmailComposer.open(parameters);
    args.set_cancel(true);
  }
  function OnReplyMailButtonClientClicking(sender, args) {
    var parameters = {
      wfId: '<%= { wf.WorkflowInstanceId } %>',
      replyToId: <Guid>,
      subject: 'Email Composer Subject',
      template: 'rat'
    }
    SEQForms.EmailComposer.open(parameters);
    args.set_cancel(true);
  }
  function OnForwardMailButtonClientClicking(sender, args) {
    var parameters = {
      wfId: '<%= { wf.WorkflowInstanceId } %>',
      forwardToId: <Guid>
    }
    SEQForms.EmailComposer.open(parameters);
    args.set_cancel(true);
  }
</script>
```

> **NOTE**
> Email Composer is supported only in workflows that are in the context of a Solution.

For general client API information on all conversation items including Email Composer, see the *Conversations View client API* section in this article.

For Email Composer, you can use the following args object parameters:

- args.detail.parameters.windowTitle: Specifies the title of the Email Composer window.
- args.detail.parameters.from: Specifies the list of email addresses a user can select from. This parameter overrides the pre-configured From email address. The first item is set as the default From address.
- args.detail.parameters.template: Applies a selected template.
- args.detail.parameters.to: Adds a list of To addresses.
- args.detail.parameters.cc: Adds a list of Cc addresses.
- args.detail.parameters.bcc: Adds a list of Bcc addresses.
- args.detail.parameters.subject: Specifies the email subject. The subject is overwritten if an email template is set.
- args.detail.parameters.body: Specifies the email body. The body is overwritten if an template is set.

> **NOTE**
> The From, To, Cc, and Bcc lists should use a comma ',' to separate between items.

Adding incoming emails to the Conversation View through built-in command

You can use the Create Conversation Item built-in-command to insert incoming emails into the Conversations View.

1. Add a Built-In-Command activity.
2. Select the category as **Conversation Operations** and command as **Create Conversation Item**.
3. Add:
   - workflowInstanceId
   - conversationItemState: should be *Incoming*, but can be any state.
4. Select the type as Email Conversation Item, and add the following command parameters:
   - From
   - To
   - CC
   - BCC
   - Subject
   - Body
     (example: IIF({EmailListenerActivityName}.Email.HasBodyHtml, {EmailListenerActivityName }.Email.PreviewBodyHtml, {EmailListenerActivityName }.Email.BodyText))
   - Attachments

> **IMPORTANT**
> To execute built-in-command that write into the Conversations View, make sure you have Execute permissions in the master workflow.

Available email states

State	Reason/Cause for the state
Draft	When a user saves an email.
InQueue	When a user send a draft or undelivered email.
Sending	When an Email Sender job fetches the email batch, and tries to send them to the email server.
Sent	When the Email Sender job succeeds to send an email.
Undelivered	When the Email Sender job fails to send an email, and received an error on email message level.
Incoming	When an email is received in the system Make sure to update through built-in-command