

# Configure Custom Conversation Type

Last Modified on 02/05/2024 5:35 am EST

Starting with V10.0, Cora SeSequence has been renamed to Cora Orchestration.

## V9.4 and later

### Overview

Cora SeSequence provides an option to extend the system conversation types, and add any custom conversation type to Cora SeSequence Solution. These types are added to the Conversations table by default, just by creating and configuring them as below.

### Creating custom conversation types

For each type, follow the steps below:

1. Create a new VS project, using the Class Library (.NET Framework).
2. Right-click the newly created project, and select NuGet Packages.  
Make sure to have the Cora SeSequence NuGets repository as a Package source.
3. Click **Browse**, and search for Conversations.
4. Install the latest version of **Genpact.CoraSeSequence.Converstaions** NuGet.
5. Add a new class to the project, and rename it with a meaningful name.
6. Set the class to inherit from the ConversationItem class.  
For example, new type called "Request", will be created in a new file, class name RequestConversationItem that inherits from ConversationItem.

```
public sealed class RequestConversationItem : ConversationItem
```

7. Add property to the newly created class:

```
public override string ItemType => <Your type name>
```

8. The base ConversationItem class, has the following default properties:
  - o CreationDate
  - o LastStateDate
  - o State
  - o Title
  - o Body
  - o WorkflowInstancelId
  - o MasterWFInstancelId
9. To add any extended property, you can add the property to the class.
  - a. Mention the following *JsonObject*:

```
[JsonObject(NamingStrategyType = typeof(CamelCaseNamingStrategy), MemberSerialization = MemberSerialization.OptIn)]  
2 references | 1 day ago | 3 authors, 7 changes  
public sealed class EmailConversationItem : ConversationItem  
{  
}
```

- b. Add values to *JsonProperty*.

```
[JsonProperty]  
1 reference | 1 day ago | 1 author, 2 changes  
public string Subject{..}
```

- c. Add *IgnoreDataMember* as value for a property that you don't want to display in the built-in-

command while creating a new instance of the conversation items.

```
[JsonProperty, IgnoreDataMember]
2 references | 1 day ago | 1 author, 2 changes
public string ForwardedFromId{...}
```

d. The *get* and *set* functions extracts and saves the properties from the propertyBag.

```
[JsonProperty]
3 references | 1 day ago | 1 author, 2 changes
public string Cc
{
    get
    {
        return GetProperty<string>(nameof(Cc));
    }
    set
    {
        PropertyBag[nameof(Cc)] = value;
    }
}
```

10. Add customer folders to the shared resources

a. Under **Components > Conversations** folder, create a new folder with the same name as the custom item.

b. In the folder add a JSON file, *metadata.json*

- itemType: full type
- name: unique type name with no spaces. Should be the same as the folder name.
- displayName: text that will appear in the view.
- supportsAttachments: determines whether the item can support attachments or not.
- supportsUICreation: determines whether the Add new action button be created for this item or not.

Example:

```
{
  "itemType": "PNMsoft.Sequence.Conversations.EmailConversationItem, PNMsoft.Sequence.Conversations, Version=9.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1",
  "name": "Email",
  "displayName": "Email",
  "supportsAttachments": true,
  "supportsUICreation": true
}
```

c. Create the following templates in the folder:

- Inline.html
- Expanded.html
- Api.js
- Styles.css

#### NOTE

In order to edit the Kendo html templates, see [this page](#).

d. Add the relevant icons to the **Images** folder.

11. Create a NuGet package, and use the package in the custom app.

12. In the nuspec file, add the following line to copy the static content to the correct location.

```
<file src="" target="tools\Shared Resources\Components\Conversations\TypeName" />
```

## Language Support

Starting from V9.5, custom conversation type supports localization.

For details, see [this article](#).

The Conversation component is translated according to form's *Globalization* settings:

- if *Localizable* = False, use the *Language* activity definition
- if *Localizable* = True, use the user's language

### Breaking change

- Starting from V10.4, when creating custom conversation types, you need to add the `view.html` file as a template for the conversation preview.
- Starting from V9.7, you need to add to the *metadata.json* file the list of all the available conversation item states.

For example, if you use the Email conversation type, you need to add the following to the *metadata.json* file.

```
"statesType": "PNMsoft.Sequence.Components.Conversations.EmailConversationItemStates,  
PNMsoft.Sequence.Components.Conversations, Version=9.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1"
```

The system uses the default state if you do not add the list of available conversation item states.