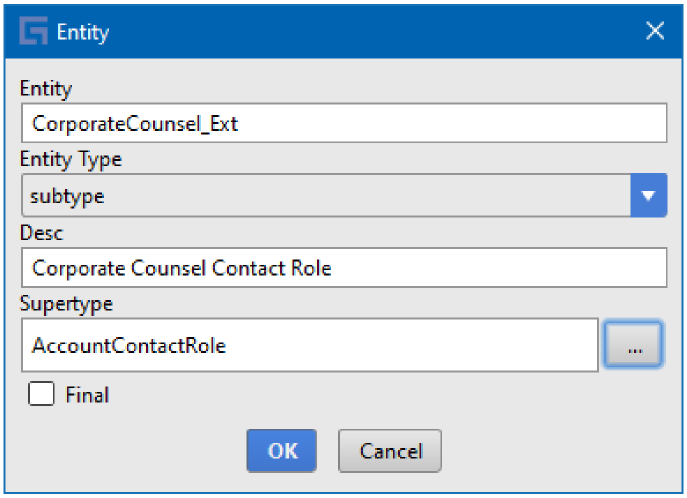
4.4 Solutions

Solution

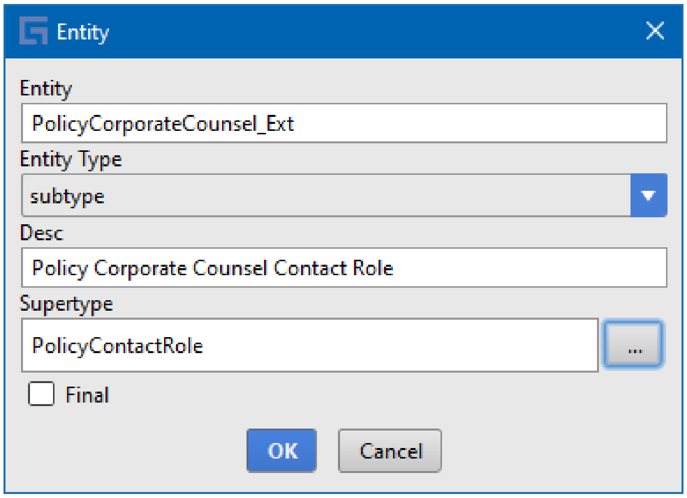
1. **Create the account contact role.**

   Create `CorporateCounsel_Ext.eti` as a subtype of `AccountContactRole`.



2. **Create the corresponding policy contact role.**

   Create `PolicyCorporateCounsel_Ext.eti` as a subtype of `PolicyContactRole`.



3. **Map policy contact role to account contact role.**

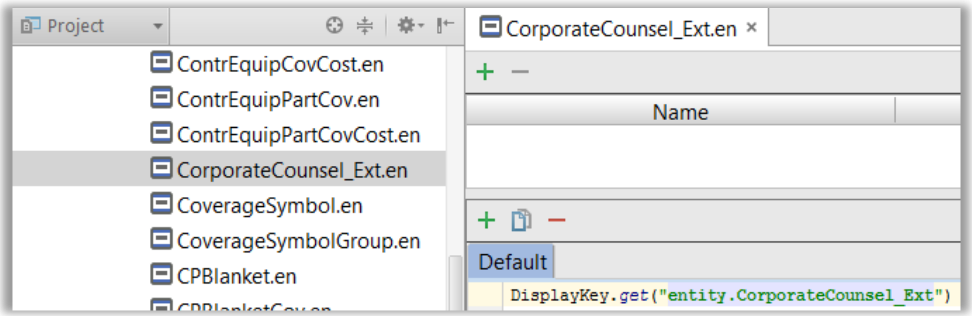
   a) Check the registered Contact Configuration Plugin in config/plugin/registry/IContactConfigPlugin.gwp.

   b) Modify the registed file, for example ContactConfigPlugin.gs or WC7ContactConfigPlugin.gs.

```
protected property get DefaultConfigs() : ContactConfig[] {
 return {

  new ContactConfig(true, {TC_COMPANY, TC_PERSON}, TC_SECONDARYCONTACT,  {}),
  new ContactConfig(true, {      TC_PERSON}, TC_CORPORATECOUNSEL_EXT,
{TC_POLICYCORPORATECOUNSEL_EXT})
 }
}
```

4. **Define the entity name and display key.**
   a) Define the entity name for the `CorporateCounsel_Ext` entity with the correct display key.
   b) If the display key does not exist, press `Alt + Enter` to create it.



5. **Re-start the server.**

4.5 References

Cookbook Recipe

Steps to add a new contact role

New contact roles can be configured by following these steps:

1. **Define new subtype of `AccountContactRole`.**
2. **Define new subtype of `PolicyContactRole`.**
3. **Map policy contact role to account contact role.**
   a) Check the registered Contact Configuration Plugin in `config/plugin/registry/IContactConfigPlugin.gwp`.
   b) Modify the registed file, for example `ContactConfigPlugin.gs` or `WC7ContactConfigPlugin.gs`.

   Syntax: `new ContactConfig(enable, {contactType}, AccountContactRole,`

   `{ PolicyContactRole1, PolicyContactRole2, … } )`

   Where ContactType is the subtype of Contact (Person and Company)

4. **Define entity name for the new contact role entity**

a) Add and reference display name as needed

5. **Modify PCFs and Gosu classes as needed**
6. **Restart server to load the new entities and entity names**

Review

Contacts

Account File Contacts page lists all contacts on an account. Policy may have some or all the Account contacts. Some contact information is shared across PolicyPeriod and some is PolicyPeriod specific. Contacts are identified by the role they fill. One contact can file multiple roles on the account and policy.

Contact role behavior patterns in PolicyCenter.

- Normal: Array of contacts connected to an entity (AdditionalNamedInsured).
- Singleton: One and only one contact connected to an entity (BillingContact).
- Simple Details: One contact with array of details (AdditionalInsured).
- Join Details: Details join contact to another entity (Driver).

Review

Strategy to configure a new contact role

- For a contact role that does not exist in base application:
  - Add a new contact role and.
  - Configure it to follow the implementation of an existing contact role.
- Compare desired behavior of contact to roles in base application and find role with most similar behavior:
  - How many of them can be created?
  - How the match attaches to other roles or objects?
- Contact role entities that are extendable include:
    - PolicyContactRole and AccountContactRole.
    - All other roles in the default configuration.
    - You can define roles for Person, Company, or Both.
- Roles can be disabled when not needed.

Review

Locations

- Locations can be created on account and reused on policy and created on policy.
- **Location Numbers**: System assigns each location a sequential number stored in `LocationNum` field.
  - `AccountLocation` and `PolicyLocation` are numbered separately.

    - Configuration developers can configure the location numbering algorithm (the end user cannot change the algorithm in the UI).
- Subtypes of `PolicyLine` entity have associated location type.
    - LOB location types have foreign key Location that points to `PolicyLocation`.
- Creating a new `AccountLoction`: To add a new `AccountLocation` to the account, call the `newLocation()` method on Account and it returns the `AccountLocation` created.
- Creating a new `PolicyLocation`: To add a new `PolicyLocation`, call the `newLocation()` method on `PolicyPeriod`.
    - `newLocation()` or
    - `newLocation(acctLoc : AccountLocation)`

      where:

      `acctLoc` is the existing `AccountLocation` to which the new policy location is associated with.

Stop