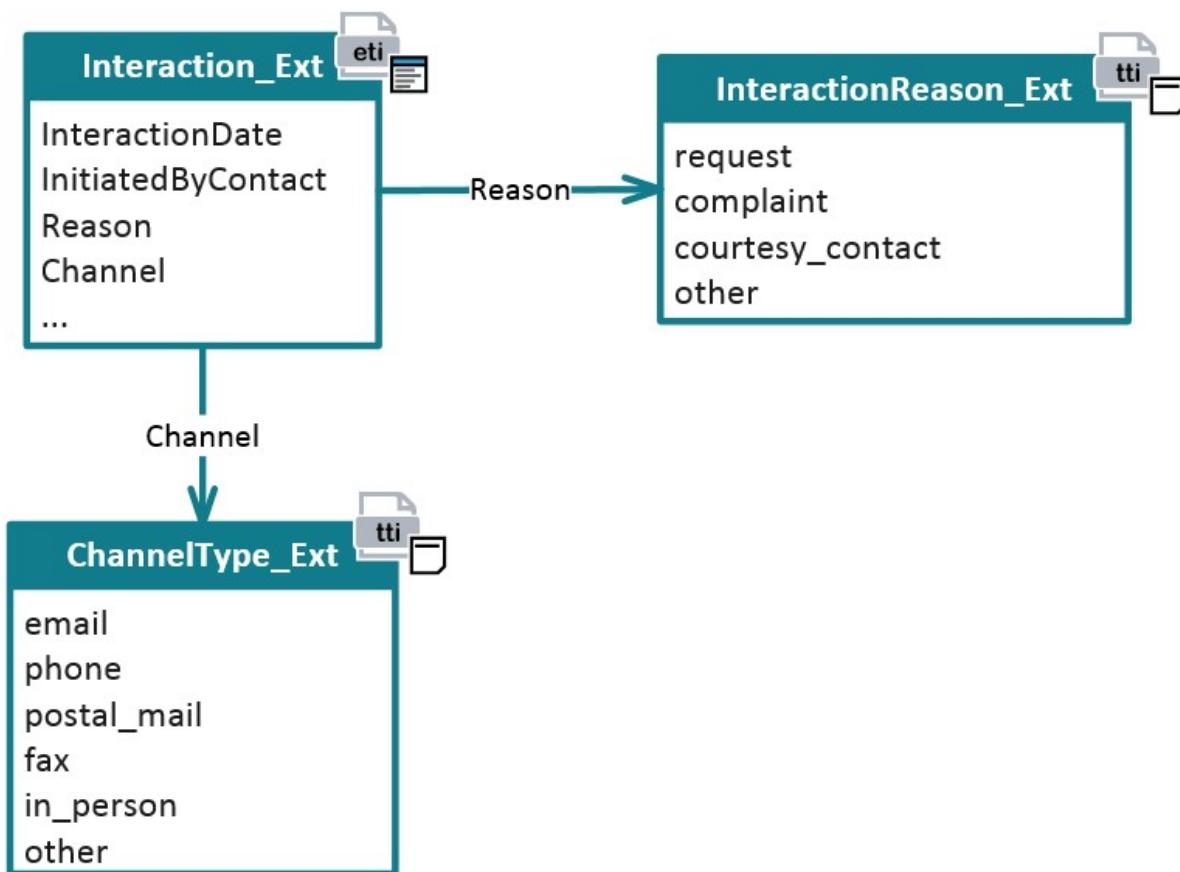


## Lesson 4 Typelists

*"After further analysis, we decided that we want to store more information about each contact interaction. Please see the diagrams below and configure the Data Model to meet the requirements." – Insurance company business analysts*



In this lab, you will use the Typelist Editor and the Entity Editor in Guidewire Studio to modify the TrainingApp data model. First, you will create two new typelists. Then, you will edit existing entities and associate the new typelists with each respective entity.

## 4.1 Prerequisites

You must first complete the following previous lesson(s):

### 6. Creating New Entities

For this lab, use TrainingApp, Guidewire Studio, and a supported web browser.

`http://localhost:8880/ab/ContactManager.do` is the default URL for TrainingApp. To view, edit, and delete various contacts, log in to TrainingApp as Alice Applegate. The login/password for Alice Applegate is `aapplegate/gw`.

## 4.2 Configuration

### 4.3 Lab: Create new typelists

As a configuration developer, you want to be able to create new typelists from scratch. In this lab, you will create and define two typelists in Guidewire Studio using the Typelist Editor. The `InteractionReason_Ext` typelist will describe the reasons for contact interactions. The `ChannelType_Ext` typelist will describe the various channels of communications for interactions.

#### 1. Create the `InteractionReason_Ext` typelist

- a) Create the new TTI file
- b) Add the following typecodes to display in the order listed using recommended naming conventions:
  - Request
  - Complaint
  - Courtesy Contact
  - Other

#### 2. Use Studio's code generation feature to process the typelist and generate the Java class

- a) If possible, use incremental code generation
- b) Verify that there were no errors during code generation
- c) Open the generated Java class and verify that you can see the newly added typecodes

### **3. Create the ChannelType\_Ext typelist**

- a) Create the new TTI file
- b) Add the following typecodes to display in the order listed using the recommended naming conventions:
  - Email
  - Phone
  - Postal Mail
  - Fax
  - In Person
  - Other

### **4. Use Studio's code generation feature to process the typelist and generate the Java class**

- a) If possible, use incremental code generation
- b) Verify that there were no errors during code generation
- c) Open the generated Java class and verify that you can see the newly added typecodes



#### **Best Practices**

Use \_Ext in the typelist name

Notice that the name of the typelist has an \_Ext suffix. For new typelists, Guidewire recommends that the name of the typelist should end with \_Ext (or start with Ext\_). This is to prevent potential conflicts during the upgrade to the next version of the Guidewire application.

## **4.4 Lab: Add typekey fields to the entity**

As a configuration developer, you want to be able to add typekey entity fields that reference typelists. In this lab, you will edit the related entities and add new typekey elements for the newly created typelists.

### **1. Edit the Interaction\_Ext entity**

- a) Add a typekey field for the InteractionReason\_Ext typelist using the recommended naming convention
- b) Add a typekey field for the ChannelType\_Ext using the recommended naming convention

### **2. Use Studio's code generation feature to generate the Java class from the entity**

- a) If possible, use incremental code generation
- b) Verify that there were no errors during code generation
- c) Open the generated Java class and verify that you can see the two new typekey fields



## Important!

Read carefully.

For each new entity element, remember to set the `nullOk` attribute to true, otherwise, the database upgrade may fail.



## Best Practices

Use `_Ext` in the entity field name

Notice that the name of the typelist has an `_Ext` suffix. For new typelists, Guidewire recommends that the name of the typelist should end with `_Ext` (or start with `Ext_`). This is to prevent potential conflicts during the upgrade to the next version of the Guidewire application.

### 4.4.1 Verification



## Activity

Verify the work you have done

As a configuration developer, you want to be able to properly deploy new and changed data model resources.

#### 1. Restart the server to deploy the changes

- a) During server restart Studio first runs the code generators, then compiles the project and finally deploys the resources

#### 2. Regenerate the Data Dictionary

- a) Follow the steps to regenerate the Data Dictionary

#### 3. Open the Data Dictionary

- a) In Windows Explorer, navigate to the data dictionary.
  - b) Open the data dictionary using your preferred browser.
4. Verify the InteractionReason\_Ext and ChannelType\_Ext typelists
  5. Verify the new typelist reference in the Interaction\_Ext entity



## 4.5 Lab Solution: Create new typelists



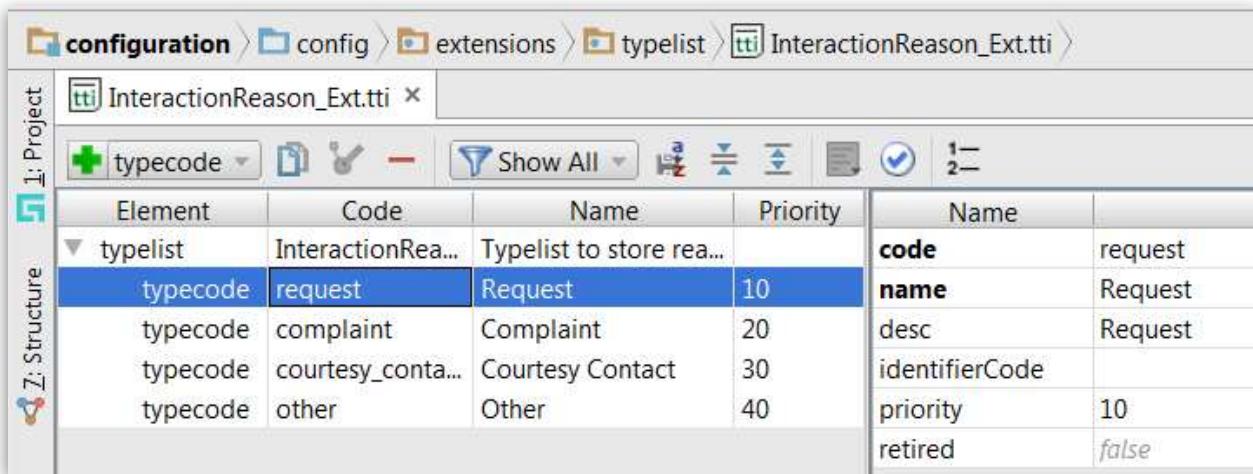
### Solution

Exact details on how to complete the lab.

1. Create the new InteractionReason\_Ext typelist



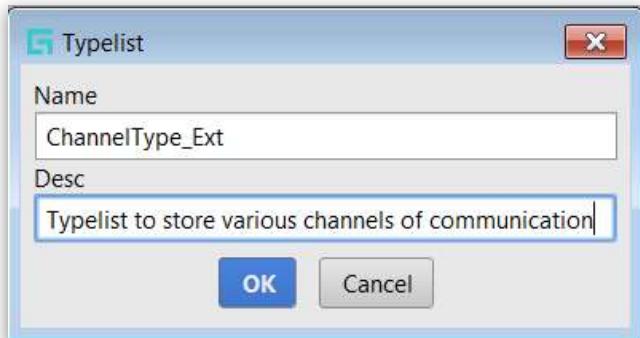
## 2. Add the typecodes



The screenshot shows a software interface for managing typecodes. The title bar indicates the path: configuration > config > extensions > typelist > InteractionReason\_Ext.tti. The main area displays a table of typecodes under the 'InteractionReason\_Ext.tti' project. The table has columns: Element, Code, Name, Priority, and several additional columns for properties like code, name, desc, identifierCode, priority, and retired.

Element	Code	Name	Priority	code	name	desc	identifierCode	priority	retired
typelist	InteractionRea...	Typelist to store rea...							
typecode	request	Request	10						
typecode	complaint	Complaint	20						
typecode	courtesy_conta...	Courtesy Contact	30						
typecode	other	Other	40						

## 3. Create the new ChannelType\_Ext typelist



#### 4. Add typecodes

The screenshot shows a software interface for managing configuration files. The top navigation bar displays the path: configuration > config > extensions > typelist > ChannelType\_Ext.tti. The left sidebar has two sections: 'Project' and 'Structure'. The 'Structure' section is currently selected and shows a tree view with a single node labeled 'typelist'. The main workspace contains a table titled 'ChannelType\_Ext.tti' with the following data:

Element	Code	Name	Priority	Name	
typelist	ChannelType_Ext.tti	Typelist to sto...		code	email
typecode	email	Email	10	name	Email
typecode	phone	Phone	20	desc	Email
typecode	postal_mail	Postal Mail	30	identifierCode	
typecode	fax	Fax	40	priority	10
typecode	inperson	In Person	50	retired	false
typecode	other	Other	60		

## 4.6 Lab Solution: Add typekey fields to an entity



### Solution

Exact details on how to complete the lab.

1. Open Interaction\_Ext.eti
2. Add a typekey field that uses the new InteractionReason\_Ext typelist

The screenshot shows the ADF Configuration tool interface. The navigation bar at the top indicates the path: configuration > config > extensions > entity > Interaction\_Ext.eti. The main area is titled "Interaction\_Ext.eti" and contains a table with columns: Element, Primary Value, Secondary Value, and Name. The table rows represent various entity attributes and their configurations. The "typekey" row is highlighted with a blue background, indicating it is the current selection. The "name" column for this row contains "Reason". The "typelist" column contains "InteractionReason\_Ext". The "nullok" column contains "true". Other rows include "Interaction\_Ext" (Primary Value: "Stores interactio...", Secondary Value: "datetime"), "InteractionDate" (Primary Value: "InteractionDate", Secondary Value: "datetime"), "InitiatedByContact" (Primary Value: "InitiatedByContact", Secondary Value: "bit"), "Summary" (Primary Value: "Summary", Secondary Value: "varchar"), "AssociatedUser" (Primary Value: "AssociatedUser", Secondary Value: "User"), "ABContact" (Primary Value: "ABContact", Secondary Value: "ABContact"), and "desc" (Primary Value: "desc").

Element	Primary Value	Secondary Value	Name
entity	Interaction_Ext	Stores interactio...	<b>name</b> Reason
column	InteractionDate	datetime	<b>typelist</b> InteractionReason_Ext
column	InitiatedByContact	bit	<b>nullok</b> true
▶ column	Summary	varchar	desc
foreignkey	AssociatedUser	User	allowInitialValueForUpgrade false
foreignkey	ABContact	ABContact	columnName
<b>typekey</b>	<b>Reason</b>	<b>InteractionReaso...</b>	createhistogram false

3. Add a typekey field that uses the new ChannelType\_Ext typelist

The screenshot shows the Oracle ADF Configuration interface. The navigation bar at the top indicates the path: configuration > config > extensions > entity > Interaction\_Ext.eti. The left sidebar has 'Project' and 'Structure' tabs, with 'Structure' currently selected. The main area is titled 'Interaction\_Ext.eti' and shows a table for the 'typekey' section. The table has columns: Element, Primary Value, Secondary Value, and Name. The 'Element' column shows various entity components like 'entity', 'column', 'foreignkey', etc. The 'Primary Value' and 'Secondary Value' columns show specific settings for each component. The 'Name' column lists configuration properties. One row is highlighted in blue, corresponding to the last entry in the table:

Element	Primary Value	Secondary Value	Name
entity	Interaction_Ext	Stores interactio...	<b>name</b>
column	InteractionDate	datetime	<b>typelist</b>
column	InitiatedByContact	bit	<b>nullok</b>
▶ column	Summary	varchar	desc
foreignkey	AssociatedUser	User	allowInitialValueForUpgrade
foreignkey	ABContact	ABContact	<i>false</i>
typekey	Reason	InteractionReaso...	columnName
typekey	Channel	ChannelType_Ext	createhistogram
typekey	Channel	ChannelType_Ext	<i>false</i>
			default