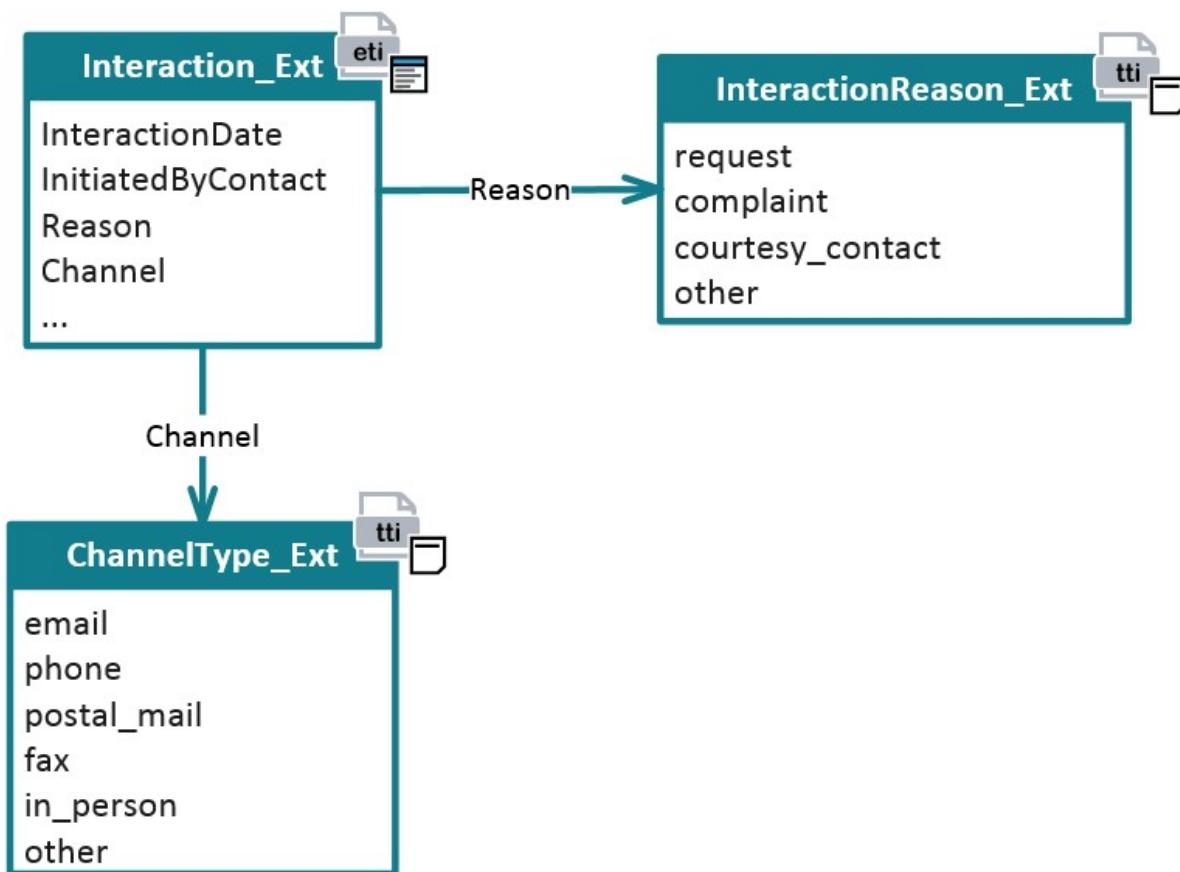


Lesson 4 Typelists

"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



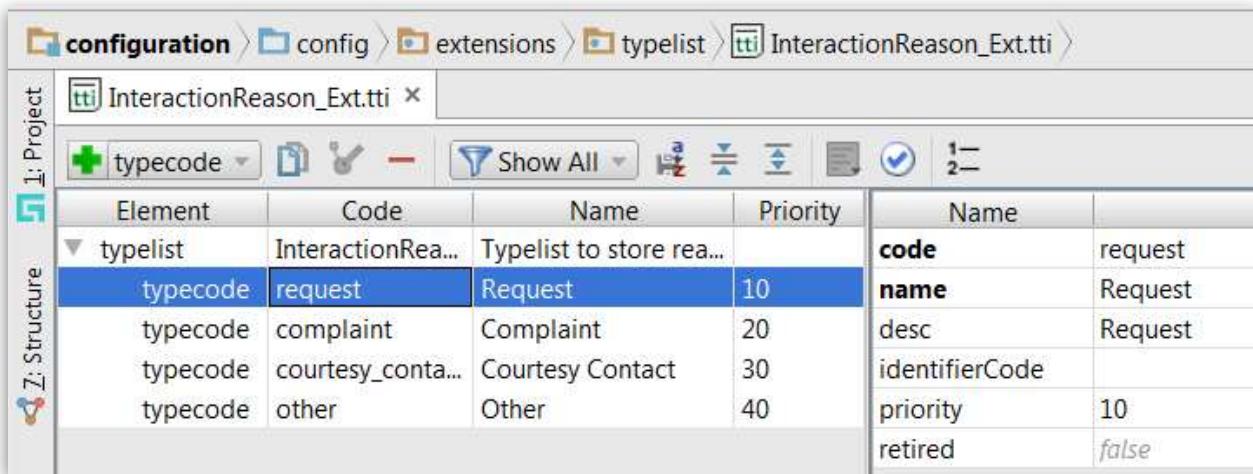
Solution

Exact details on how to complete the lab.

1. Create the new InteractionReason_Ext typelist



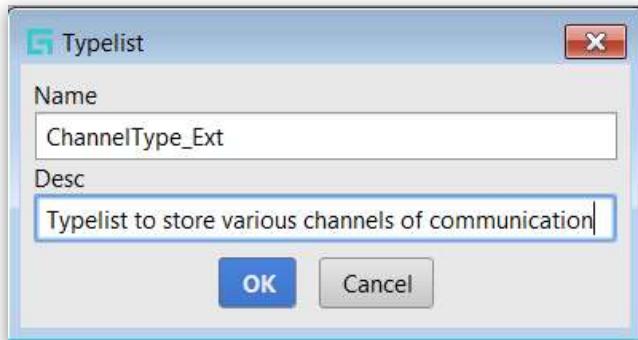
2. Add the typecodes



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: '1: Project' and '2: Structure'. The '2: Structure' section is currently active, showing a tree view where 'typelist' is expanded to show its children: 'typecode email', 'typecode phone', 'typecode postal_mail', 'typecode fax', 'typecode inperson', and 'typecode other'. Below this tree is a table with the following data:

Element	Code	Name	Priority	Name	
typelist	ChannelType_...	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		