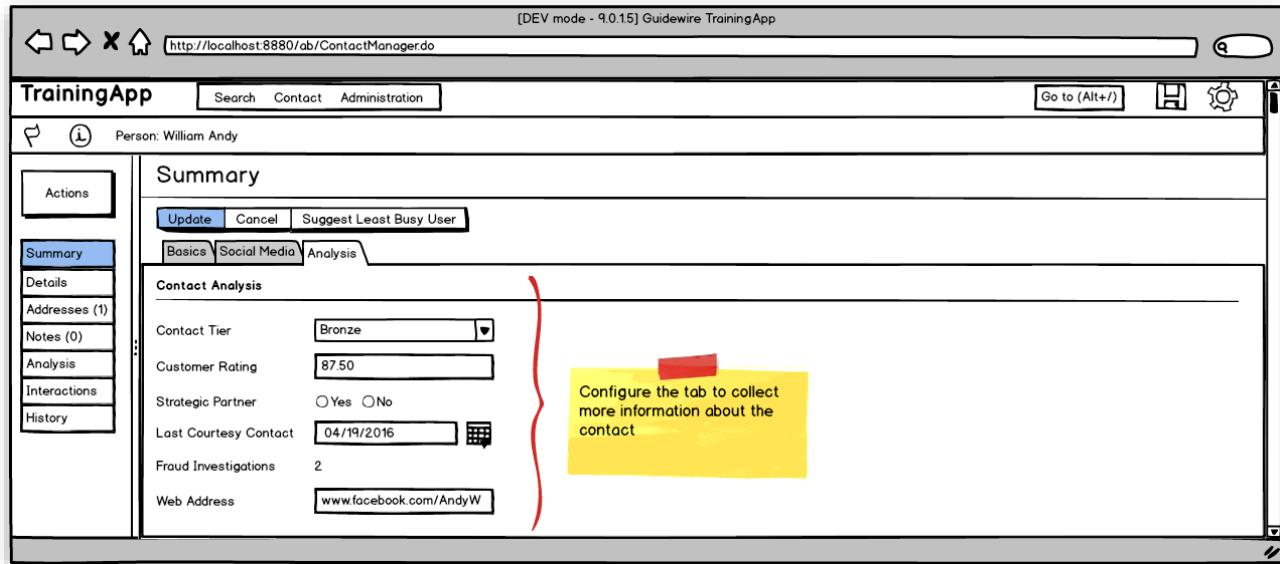


# Lesson 3 Extending the Data Model

An insurance company wants to extend TrainingApp functionality by capturing more details about each contact. The complete requirement will be implemented over multiple modules.



In this exercise your job is to make the necessary **data model changes** to meet customer requirements. As a configuration developer, you will use the Entity Editor in Guidewire Studio to modify the TrainingApp data model. You will implement the **user interface changes** later in a later lesson.

## 3.1 Prerequisites

For this exercise, 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.

## 3.2 Lab: Modify an existing Entity Extension

You will use the Entity Editor to modify the existing ABContact.etx Entity Extension and use the column element to add new data fields to the ABContact base application entity.

Review the field definitions received from the business analysts. This table below summarizes the requirements for each field. Note: The Contact Tier field is missing from this list, because it has already been added by one of our fellow configuration developers.

Field name	Datatype	Null ok?
WebAddress_Ext	String of up to 40 characters	true
FraudInvestigationNum_Ext	Integer	true
LastCourtesyContact_Ext	Date (a date and time value)	true
CustomerRating_Ext	A decimal in the format XXX.Y (Values will range from 0.0 to 999.9)	true
IsStrategicPartner_Ext	Bit (a Boolean value)	true



## Important!

Read carefully.

For each new entity element, remember to set the nullok attribute to true if not specifically defined to be false. When required, add an element description and specify column parameters.



## Best practice



Use \_Ext in entity field name

Notice that every field in the table above has an \_Ext suffix. For fields that are added to a base application entity, Guidewire recommends that the field name should end with \_Ext. This is to prevent potential conflicts during the upgrade to the next version of the Guidewire application.

### 1. Open Guidewire Studio

- From the TrainingApp folder, open a command window and start Guidewire Studio for TrainingApp.
- Note: Alternatively, use the Start TrainingApp Studio shortcut.

### 2. Navigate to ABContact.etx

**3. Add fields to ABContact.etx to capture additional details**

- a) Add the elements defined in the table below.

Field name	Datatype	Null ok?
WebAddress_Ext	String of up to 40 characters	True
FraudInvestigationNum_Ext	Integer	True
LastCourtesyContact_Ext	Date (a date and time value)	True
CustomerRating_Ext	A decimal in the format XXX.Y (Values will range from 0.0 to 999.9)	True
IsStrategicPartner_Ext	Bit (aBoolean value)	True

**4. Validate your changes in Guidewire Studio and generate the Java class**

- Either click on the validate icon in the Entity Editor or use CTRL + S to save your changes
- If the code generation was successful (no errors shown in the Codegen tool window) then open the generated Java class and verify you can find your newly added fields

**5. Deploy your changes**

- From studio, restart the server using Debug 'Server'.
- If open, review the Messages Make window for compilation errors.
- Review the Debug console for errors.
- Verify that the application is running in the Debug console.

### 3.3 Lab: Extend a base application Typelist

*"We also defined new customer service tiers, so we can properly prioritize customer request. Please see the diagram below and configure the Data Model to meet the requirement." – Insurance company business analysts*



As a configuration developer, you want to be able to extend base application Typelists so you can add new Typecodes as needed. In this exercise, you will create an extension for the CustomerServiceTier Typelist and use the Typelist Editor to define three new Typecodes.

### 3.3.1 Configuration

#### 1. Extend the CustomerServiceTier Typelist

- a) Create a new TTX file
- b) Define the following Typecodes to display in the order listed using recommended naming conventions:
  - Platinum
  - Gold
  - Silver

#### 2. Validate your changes in Guidewire Studio and generate the Java class

- a) Either click on the validate icon in the Typelist Editor or use CTRL + S to save your changes
- b) If the code generation was successful (no errors shown in the Codegen tool window) then open the generated Java class and verify you can find your newly added Typecodes

#### 3. Deploy your changes

- a) From studio, restart the server using Debug 'Server'.
- b) If open, review the Messages Make window for compilation errors.
- c) Review the Debug console for errors.
- d) Verify that the application is running in the Debug console.



#### Best practice

Use intervals when defining priorities

Guidewire recommends using intervals when defining priorities. For example.: 10, 20, 30 This enables you to insert a new value between two existing values without renumbering the other priorities. E.g. you can insert a new Typecode in a later release with the priority 15



## Best Practice



Naming typecodes

For typecodes that are added to a base application typelist, Guidewire recommends that the typecode's code should end with \_Ext. This is to prevent potential conflicts during the upgrade to the next version of the Guidewire application.

### 3.3.2 Verification



## Activity

Verify the work you have done

Regenerating the data dictionary is not required. Regenerating the data dictionary is a best practice because it creates current documentation for the data model. In addition, the build process for the data dictionary can identify issues beyond schema validation such as referential integrity in the data model.

#### 1. Build the data dictionary from the command window

- a) From the bin folder, open a command window.
- b) In the command window, enter the command to build the data dictionary.

#### 2. Open the data dictionary

- a) In Windows Explorer, navigate to the data dictionary.
- b) Open the data dictionary using your preferred browser.

#### 3. View the ABContact entity

- a) Verify each new field and associated datatype.

#### 4. View the CustomerServiceTier Typelist

- a) Verify each new Typecodes.



## Best practice

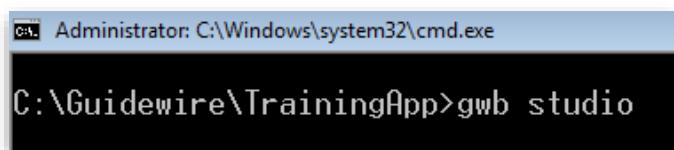
Regenerate the Data Dictionary

Regenerating the data dictionary is not required. Regenerating the data dictionary is a best practice because it creates current documentation for the data model. In addition, the build process for the data dictionary can identify issues beyond schema validation such as referential integrity in the data model.

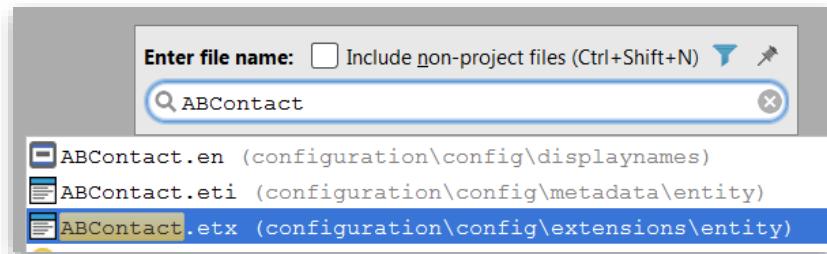
## 3.4 Lab Solution: Modify an existing Entity Extension



### 1. Open Guidewire Studio



### 2. Use CTRL + SHIFT + N to navigate to the existing Entity Extension



### 3. Add the new elements (and subelements) to ABContact.etx

#### 4. Add the new WebAddress\_Ext column

- Right click on the WebAddress\_Ext column and add select Add New... columnParam to add a columnParam with the following details:

A screenshot of the ABContact.etx configuration editor. The left pane shows a tree structure with "column" selected. The right pane displays two tables: one for columns and one for column parameters. In the columns table, there is a row for "WebAddress\_Ext" with "varchar" as the primary value and "40" as the size. In the column parameters table, there are two rows: one for "name" with "size" as the value, and one for "value" with "40" as the value.

### 5. Add the new FraudInvestigationNum\_Ext integer column:

ABContact.etx

The screenshot shows the ABContact.etx configuration window. On the left, there's a tree view with 'column' selected. Under 'column', there are three rows: 'WebAddress\_Ext' (Primary Value: varchar, Secondary Value: varchar), 'FraudInvestigationN...' (Primary Value: integer, Secondary Value: integer), and 'implementsInter...' (Primary Value: com.guidewire.pl.do..., Secondary Value: com.guidewire.ab.d...). On the right, there's a table titled 'ABContact.eti' with four rows: 'name' (Value: FraudInvestigationNum\_Ext), 'type' (Value: integer), 'nullok' (Value: true), and 'desc' (Value: empty).

Element	Primary Value	Secondary Value	Name	Value
column	WebAddress_Ext	varchar	<b>name</b>	FraudInvestigationNum_Ext
columnParam	size	40	<b>type</b>	integer
column	FraudInvestigationN...	integer	<b>nullok</b>	true
implementsInter	com.guidewire.pl.do...	com.guidewire.ab.d...	<b>desc</b>	

#### 6. Add a new LastCourtesyContact\_Ext column:

ABContact.etx

The screenshot shows the ABContact.etx configuration window. On the left, there's a tree view with 'column' selected. Under 'column', there are four rows: 'WebAddress\_Ext' (Primary Value: varchar, Secondary Value: varchar), 'FraudInvestigationN...' (Primary Value: integer, Secondary Value: integer), 'LastCourtesyContact...' (Primary Value: datetime, Secondary Value: datetime), and 'implementsInter...' (Primary Value: com.guidewire.pl.do..., Secondary Value: com.guidewire.ab.d...). On the right, there's a table titled 'ABContact.eti' with four rows: 'name' (Value: LastCourtesyContact\_Ext), 'type' (Value: datetime), 'nullok' (Value: true), and 'desc' (Value: Date and Time of courtesy interaction).

Element	Primary Value	Secondary Value	Name	Value
▶ column	WebAddress_Ext	varchar	<b>name</b>	LastCourtesyContact_Ext
column	FraudInvestigationN...	integer	<b>type</b>	datetime
column	LastCourtesyContact...	datetime	<b>nullok</b>	true
implementsInter	com.guidewire.pl.do...	com.guidewire.ab.d...	<b>desc</b>	Date and Time of courtesy interaction

#### 7. Add a new decimal column with the following details:

ABContact.etx

The screenshot shows the ABContact.etx configuration window. On the left, there's a tree view with 'column' selected. Under 'column', there are four rows: 'LastCourtesyContact...' (Primary Value: datetime, Secondary Value: datetime), 'CustomerRating\_Ext' (Primary Value: decimal, Secondary Value: decimal), and two 'implementsInter...' rows. On the right, there's a table titled 'ABContact.eti' with five rows: 'name' (Value: CustomerRating\_Ext), 'type' (Value: decimal), 'nullok' (Value: true), and 'desc' (Value: Customer Rating). The 'implementsInter...' row is also present in the table.

Element	Primary Value	Secondary Value	Name	Value
column	LastCourtesyContact...	datetime	<b>name</b>	CustomerRating_Ext
column	CustomerRating_Ext	decimal	<b>type</b>	decimal
implementsInter	com.guidewire.pl.do...	com.guidewire.ab.d...	<b>nullok</b>	true
implementsInter	com.guidewire.pl.sy...	com.guidewire.ab.d...	<b>desc</b>	Customer Rating

#### 8. Right click on the CustomerRating\_Ext column and add select Add New... columnParam to add a columnParam with the following details:

ABContact.etx

The screenshot shows the ABContact.etx extension in the Studio interface. On the left, there's a tree view with 'columnParam' selected. In the main pane, there are two tables. The first table has three columns: Element, Primary Value, and Secondary Value. It contains rows for 'column' (Primary Value: LastCourtesyConta..., Secondary Value: datetime) and 'columnParam' (Primary Value: CustomerRating\_Ext, Secondary Value: decimal). The second table has two columns: Name and Value. It contains rows for 'name' (Value: precision) and 'value' (Value: 4). The 'precision' row in the first table is highlighted.

Element	Primary Value	Secondary Value	Name	Value
column	LastCourtesyConta...	datetime	name	precision
column	CustomerRating_Ext	decimal	value	4
columnParam	precision	4		

Name	Value
name	precision
value	4

9. Right click on the CustomerRating\_Ext column again and add select Add New... columnParam to add a second columnParam with the following details:

Screenshot of the ABContact.etx extension, with the new column

ABContact.etx

The screenshot shows the ABContact.etx extension in the Studio interface. The configuration is identical to the previous one, except for the addition of a new 'columnParam' row in the first table. This new row has 'columnParam' as its element, 'scale' as its primary value, and 1 as its secondary value. The second table remains the same.

Element	Primary Value	Secondary Value	Name	Value
column	LastCourtesyConta...	datetime	name	scale
column	CustomerRating_Ext	decimal	value	1
columnParam	precision	4		
columnParam	scale	1		

10. Add the new IsStrategicPartner\_Ext column with the following details:

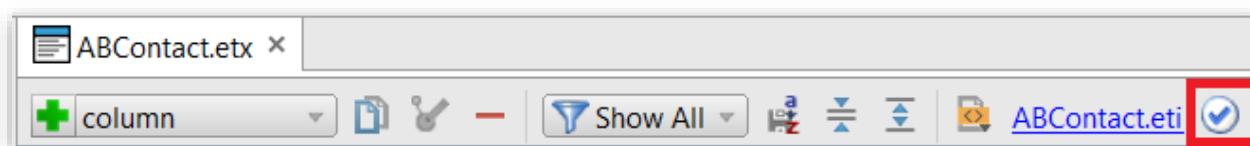
ABContact.etx

The screenshot shows the ABContact.etx extension in the Studio interface. The configuration table has a new row for the 'column' element. This row has 'IsStrategicPartner\_Ext' as its primary value and 'bit' as its secondary value. The second table also has a new row for 'name' with the value 'IsStrategicPartner\_Ext'. Other rows in both tables remain the same.

Element	Primary Value	Secondary Value	Name	Value
column	IsStrategicPartner_Ext	bit	name	IsStrategicPartner_Ext
implementsInterface	com.guidewire.pl.d...	com.guidewire.ab....	type	bit
implementsInterface	com.guidewire.pl.s...	com.guidewire.ab....	nullok	true
implementsInterface	gw.ab.addressbook...	com.guidewire.ab....	desc	Is Strategic Partner?

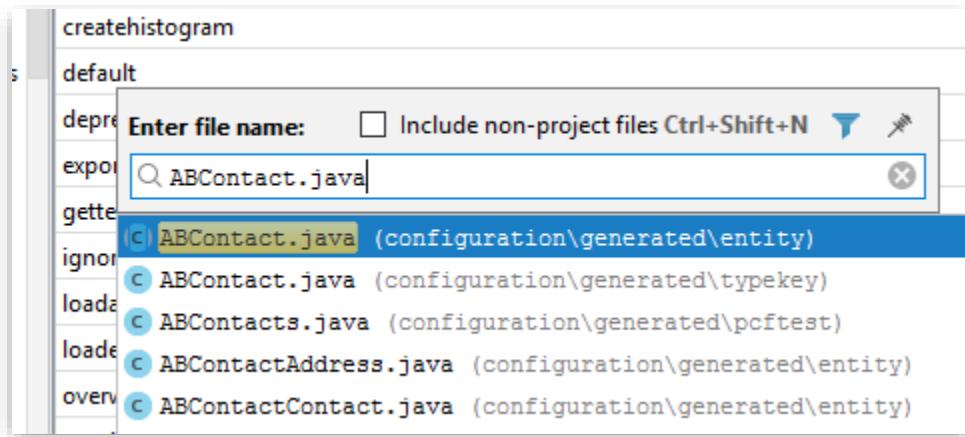
11. Validate changes in Studio and generate the Java class

- a) Either click on the validate icon in the Entity Editor or use CTRL + S to save your changes

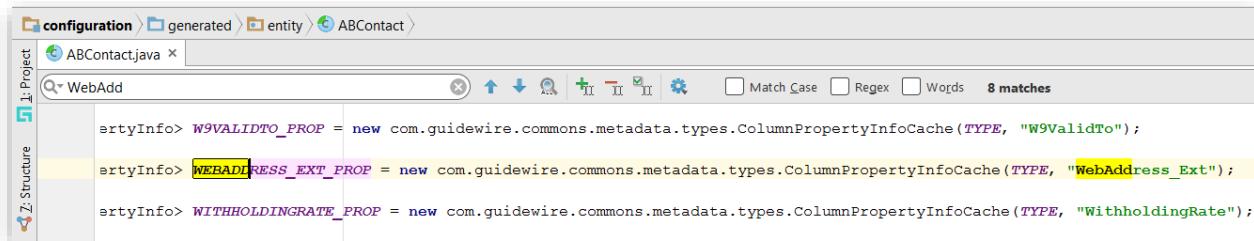


**12. If the code generation was successful (no errors shown in the codegen tool window) then open the generated Java class and verify you can find your newly added fields.**

- Use CTRL + SHIFT + N to search for the Java class. Note that the Java class is in the configuration\generated\entity package.



The simplest way to locate the new properties is searching for them using the **CTRL + F** shortcut. For example, we are searching for WebAddress\_Ext in the following screenshot:

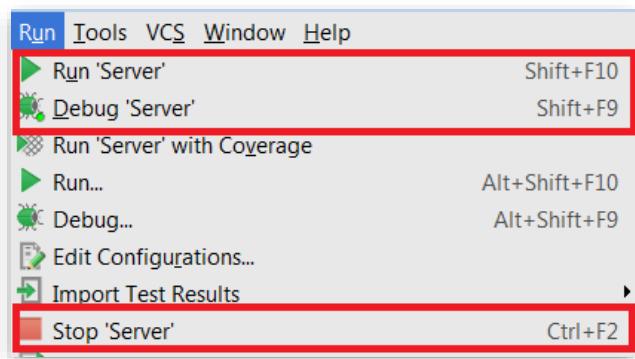


### **13. Build the Data Dictionary**

A screenshot of a Windows Command Prompt window titled "Administrator: C:\windows\system32\cmd.exe". The command "C:\Guidewire\TrainingApp>gwb genDataDictionary" is being typed into the prompt.

### **14. Deploy changes**

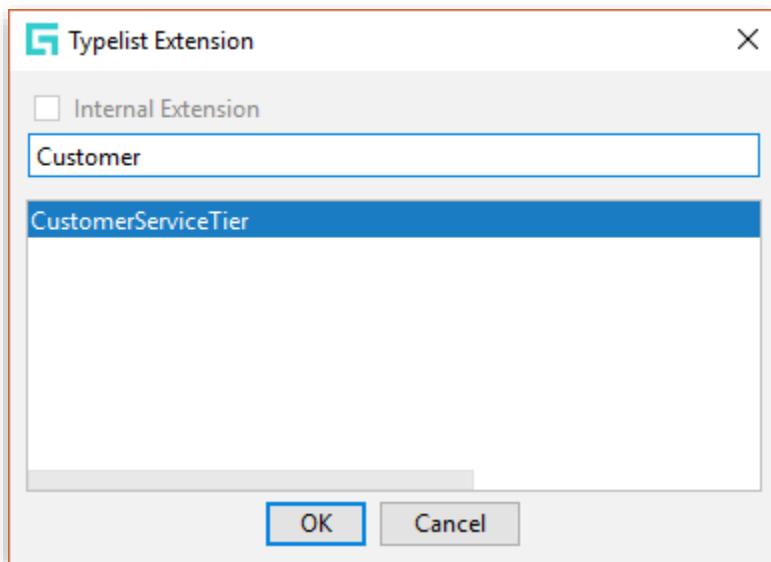
- To restart the server, first select Main Menu → Run → Stop, then select Main Menu → Run → Run 'Server' or Debug 'Server'.



### 3.5 Lab Solution: Extend a base application Typelist



#### 1. Create the Typelist Extension



2. Add Typecodes by clicking the typecode button (the icon with the green plus icon)

The screenshot shows the configuration interface with the path: configuration > config > extensions > typelist > CustomerServiceTier.ttx. A new typecode named 'platinum\_Ext' is being added. The 'Element' column lists 'typecode platinum\_Ext'. The 'Code' column shows 'platinum\_Ext'. The 'Name' column is 'Platinum' and the 'Priority' column is '10'. To the right, a detailed view of the typecode properties is shown:

Name	Value
<b>code</b>	platinum_Ext
<b>name</b>	Platinum
desc	Platinum
identifierCode	
priority	10
retired	false

The screenshot shows the configuration interface with the path: configuration > config > extensions > typelist > CustomerServiceTier.ttx. A new typecode named 'gold\_Ext' is being added. The 'Element' column lists 'typecode gold\_Ext'. The 'Code' column shows 'gold\_Ext'. The 'Name' column is 'Gold' and the 'Priority' column is '20'. To the right, a detailed view of the typecode properties is shown:

Name	Value
<b>code</b>	gold_Ext
<b>name</b>	Gold
desc	Gold
identifierCode	
priority	20
retired	false

The screenshot shows the configuration interface with the path: configuration > config > extensions > typelist > CustomerServiceTier.ttx. A new typecode named 'silver\_Ext' is being added. The 'Element' column lists 'typecode silver\_Ext'. The 'Code' column shows 'silver\_Ext'. The 'Name' column is 'Silver' and the 'Priority' column is '30'. To the right, a detailed view of the typecode properties is shown:

Name	Value
<b>code</b>	silver_Ext
<b>name</b>	Silver
desc	Silver
identifierCode	
priority	30
retired	false