

Policy Center

Lesson Outline

- Data Model
- Extending Entities
- Creating Entities
- Creating Typelist
- Adding Typefilter
- Data Dictionary

Data Model

Configuration

3 Tier Architecture

- Data Tier – Data Model Entity
- Presentation Tier – PCF file
- Application Tier – Gosu Class

Data Tier

- Data Model - set of data objects and information about their relationships
- Every Guidewire data model includes :
 - Entity
 - Entity Field
 - Typelist

Entity

- Business objects stored in its own database table. Eg : ABContact, ABUser

ABContact

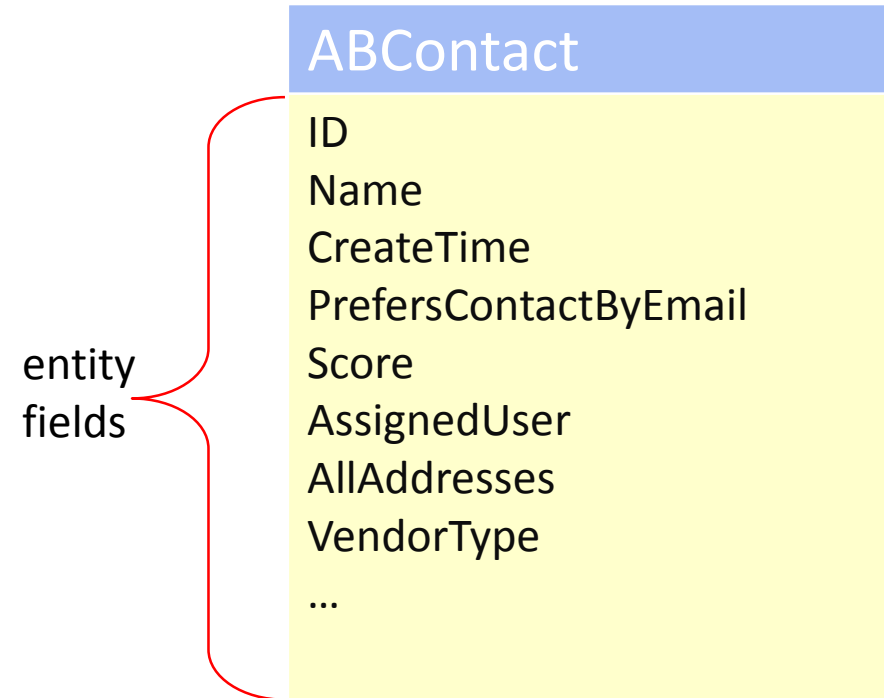
ID
Name
CreateTime
PrefersContactByEmail
Score
AssignedUser
AllAddresses
VendorType
...

Table - dbo.ab_abcontact*

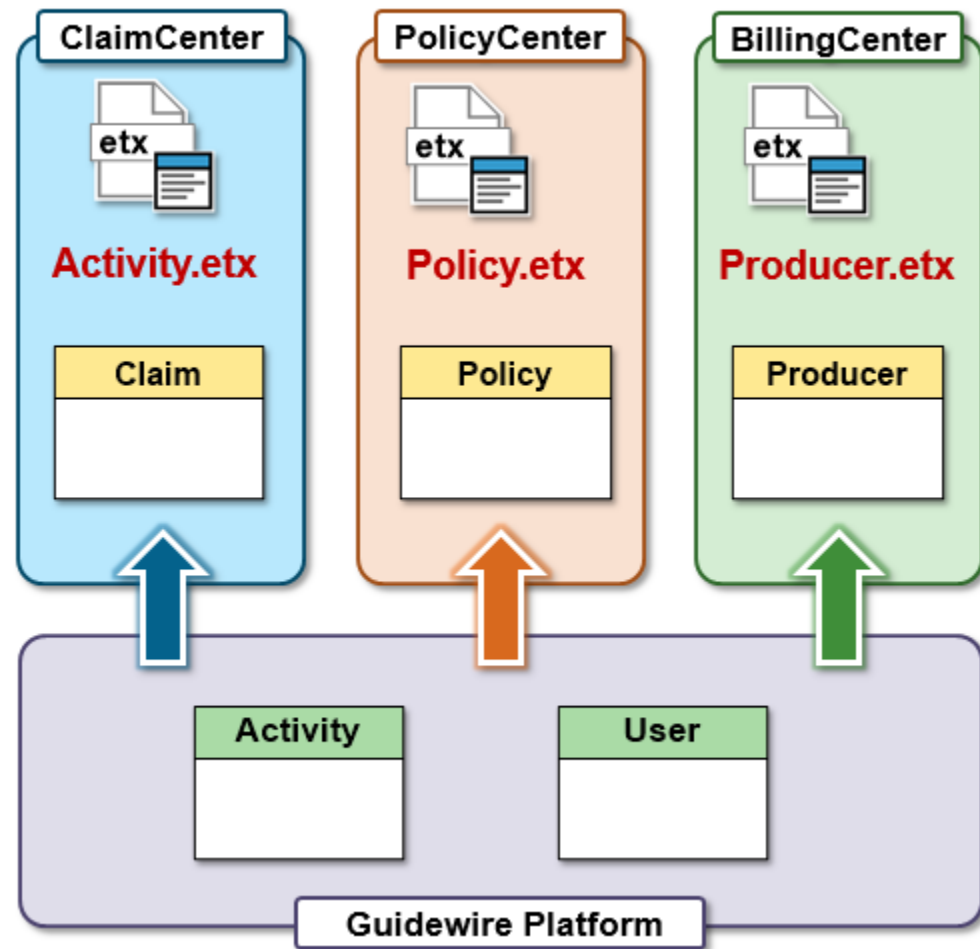
	ID	Name	CreateTime	PrefersContact...	Score	AssignedUser
	64	United Natural Foods Inc	9/4/2009 1:51:32 PM	False	NULL	2
	65	3M	9/4/2009 1:51:32 PM	False	NULL	3
	74	Express Auto	9/4/2009 1:51:32 PM	False	81	3
	75	European Autoworks	9/4/2009 1:51:32 PM	True	NULL	4
	76	M B Garage	9/4/2009 1:51:32 PM	False	NULL	7
	77	Burlingame Saab	9/4/2009 1:51:32 PM	False	NULL	2
	78	Menlo Park Chevron	9/4/2009 1:51:32 PM	True	72	6
	79	Cupertino's Smog Pro and Auto Repair	9/4/2009 1:51:32 PM	False	NULL	2
	80	Meinecke Car Care Center	9/4/2009 1:51:32 PM	False	NULL	2
	81	Morgan Hill Auto Body	9/4/2009 1:51:32 PM	False	NULL	2
	82	Hollister Muffler and Quick Lube	9/4/2009 1:51:32 PM	True	NULL	2

Entity Fields

- Value (or set of values) used to define the state or nature of a specific instance of the entity. Eg : ABContact's Name field
- Four general types of entity fields
 - Data key – to store data
 - Foreign key – to refer entity
 - Array key – to refer array of entity
 - Type key – to refer typelist



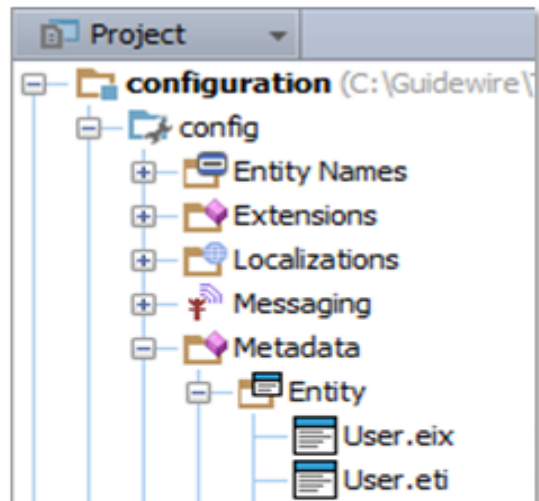
Layers of Entity



- Platform-level entities “eti” are common to all Guidewire applications
- Application-level entities “eix” are specific to given application
- Entities extensions “etx” are addons to the existing entities

Project View

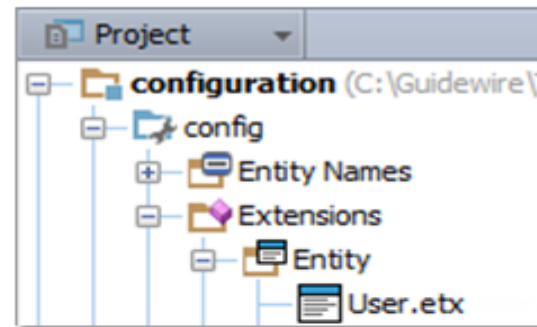
...\metadata\entity\



- Read-only files
 - Entity (ETI)
 - Internal entity extension (EIX)



...\extensions\entity



- Editable files
 - Entity (ETI)
 - Entity extension (ETX)

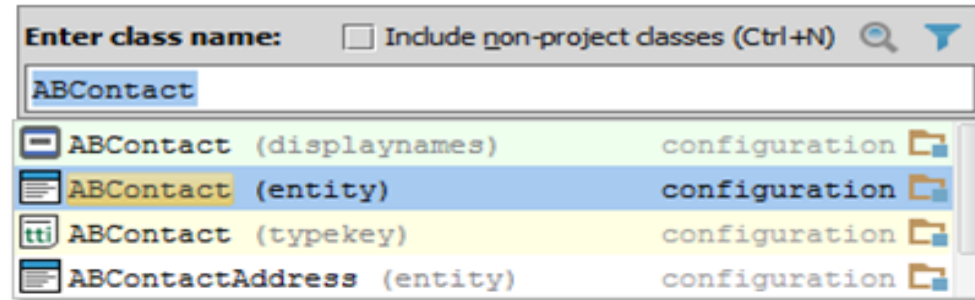
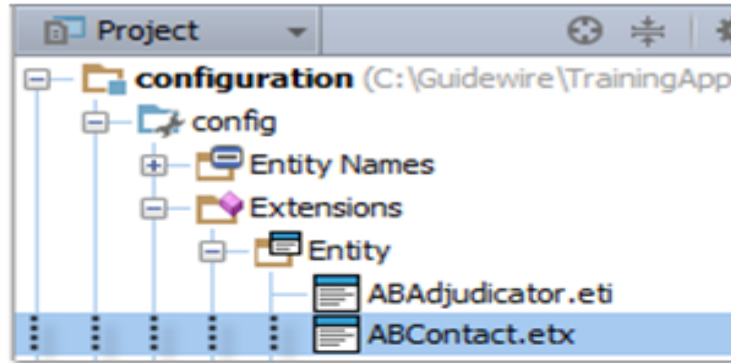


Extending Entity

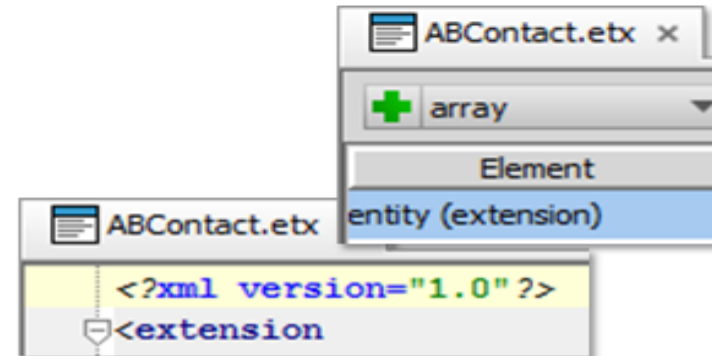
Steps to edit an entity extension

1. Navigate to the entity extension
2. Add elements and specify attribute values
3. Optionally regenerate the dictionary
4. Deploy the extension entity

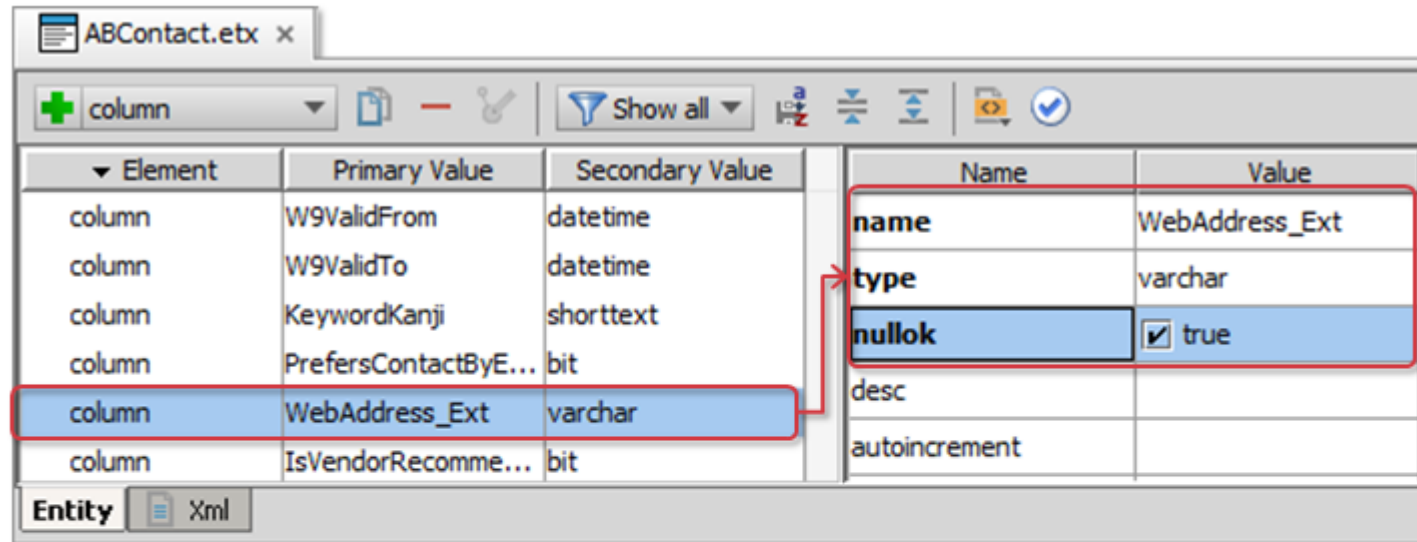
Step 1: Navigate to the entity extension



- Navigate to an entity in ...\Extensions\Entity\
 - Project View or using CTRL+N
- Verify that you have selected an entity extension file (ETX)
 - Top element reads entity (extension)
 - XML is <extension />



Step 2: Add elements and define attributes



- Toolbar to add an element for a field, e.g., `<column />`
- Define element attributes
 - Name is the name of the field; Use `_Ext` for field name ending
 - Type is the data type
 - Nullok defaults to false, so set to true in most cases

Step 3: Optionally regenerate dictionary

- **gwXX regen-dictionary**
- Process builds entire entity model including base and custom entities
- Identifies errors in the data model beyond Entity Editor schema validation

```
C:\Guidewire\TrainingApp\bin>gwta regen-dictionary
regen-entity-model-xml:
=====
= Running main class:
  com.guidewire.tools.dictionary.data.EntityModelXmlTool
    [java] --- Guidewire Entity Model In Xml ---
...
ERROR Errors found in ABContact
ERROR Attribute 'value' in element 'columnParam' is required.
ERROR ColumnIsValid - The column "WebAddress_Ext" on entity
  "ABContact" declares an invalid data type, "varchar". null
```

Step 4: Deploy the entity extension

Restart Server

- Entity Extension

- bin command window
 - **gwXX dev-stop**
 - **gwXX dev-start**



Task – Extend Entity

- Create extension BACost.etx to capture the following information.
 - Currency_Ext – varchar of 20 characters
 - Fees_Ext - integer

Creating Entity

Steps to create a custom entity

1. Create the entity file
2. Add elements and specify attribute values
3. Optionally regenerate the dictionary
4. Deploy the custom entity

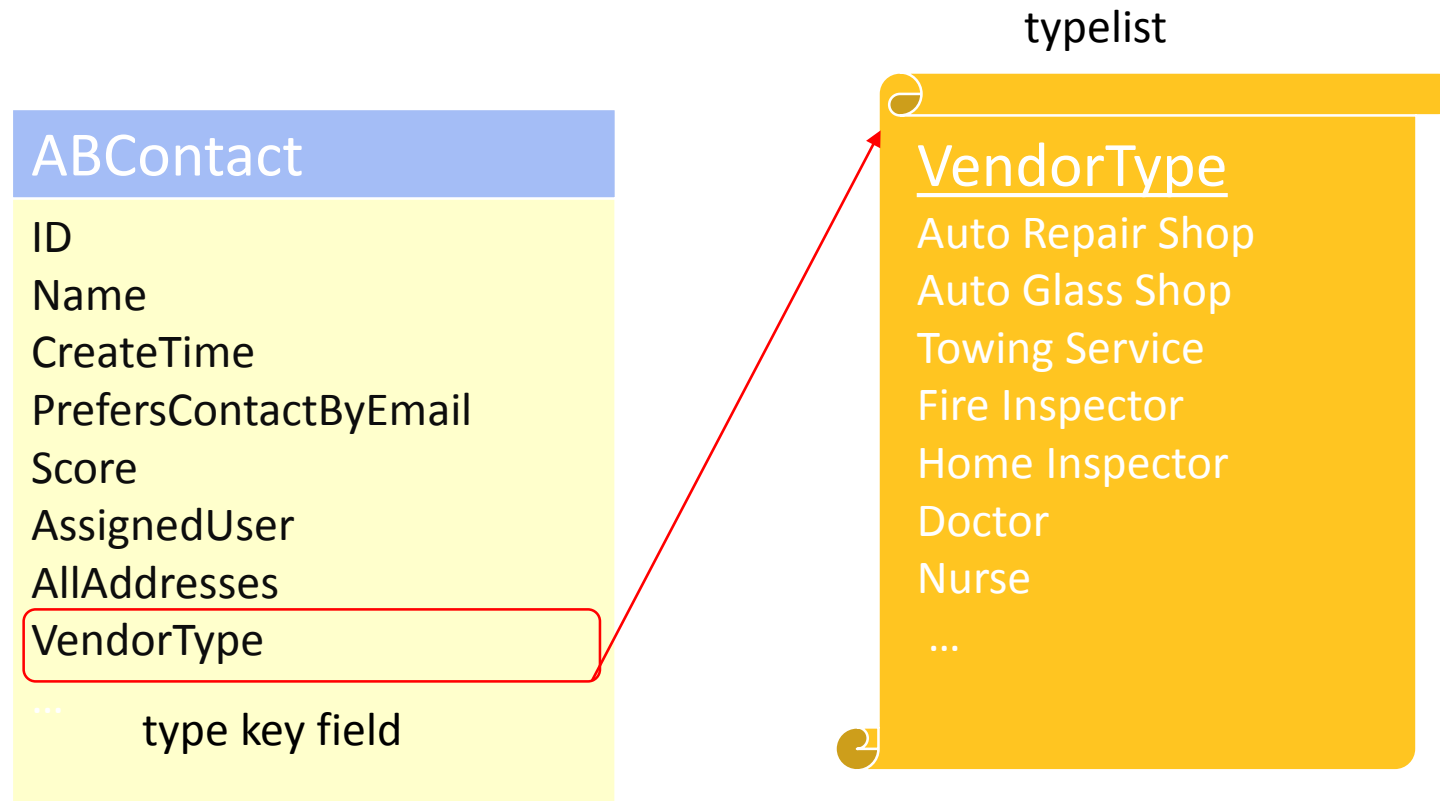
Task – Create Entity

- Create a new entity named HouseDetails_Ext.eti and add the following elements :
 - HouseNumber – integer
 - HouseName – varchar
 - FoundationDate - dateonly
- Create a new entity named RouteDetails_Ext.eti and add following elements :
 - RoadName – varchar
 - OfficeDistance – integer
 - BusFare – money
- Add a foreign key from PolicyPeriod to HouseDetails_Ext
- Add an array key from HouseDetails_Ext to RouteDetails_Ext

Typelist

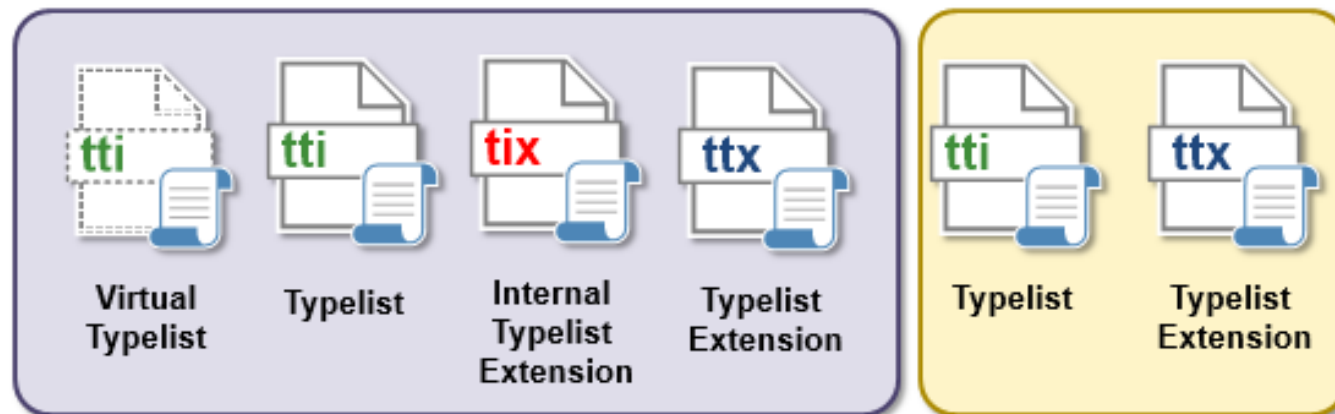
Typelist

- Predefined list of values that constrains a field



Kinds of Typelist

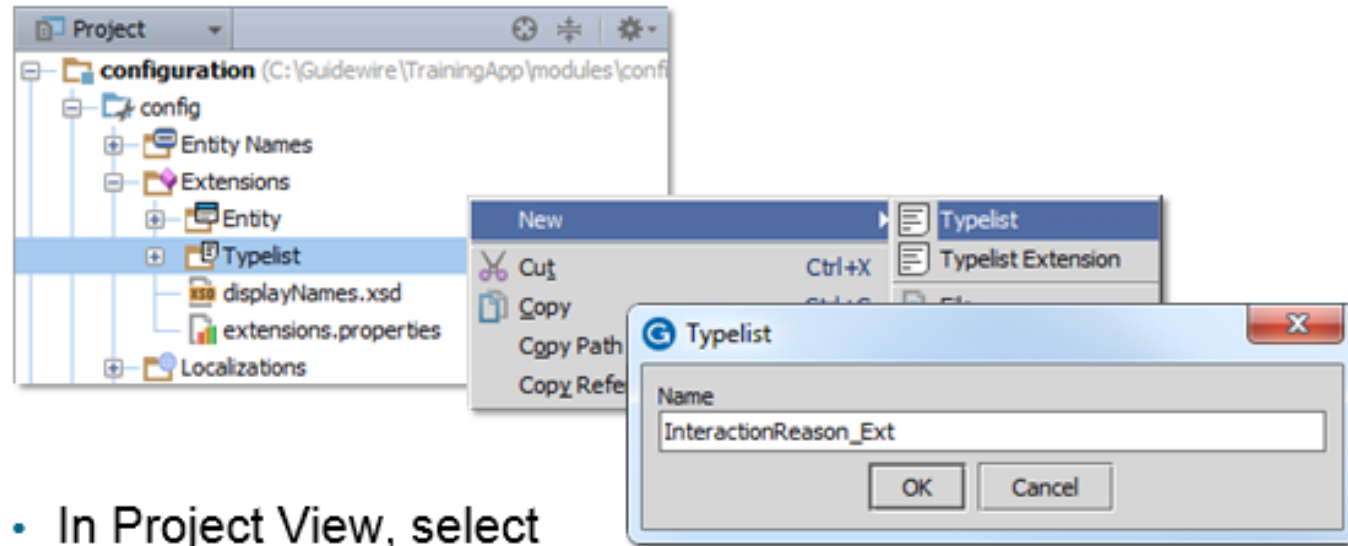
- **Internal**
 - Internal base application logic
 - Cannot modify or extend
 - Read-only
- **Extendable**
 - Base application
 - Can modify typelist or typelist extension
- **Custom**
 - Customer creates as part of custom configuration



Steps to create a typelist

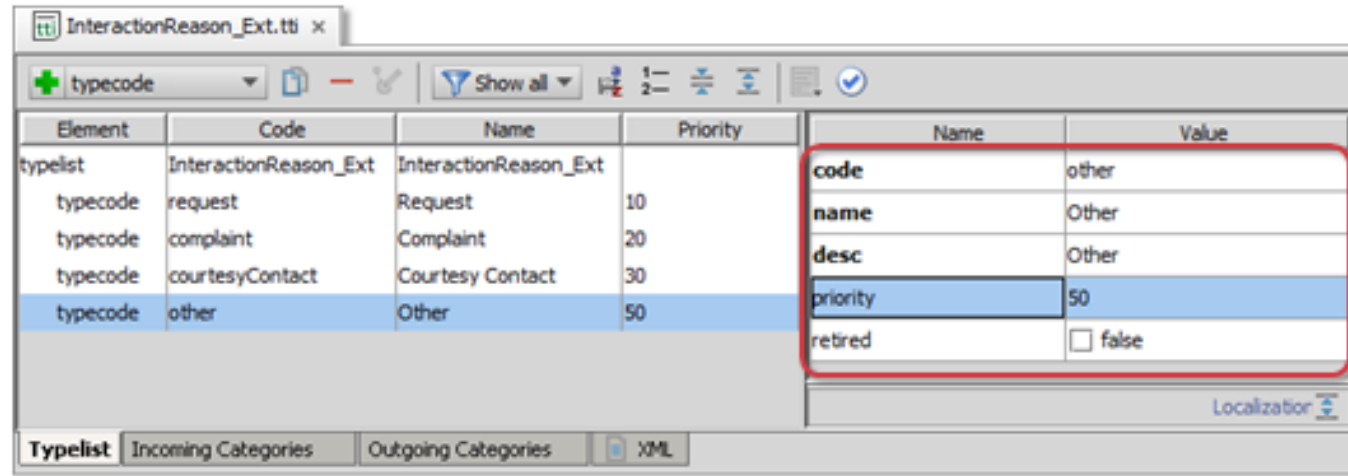
1. Create the typelist file
2. Define typecodes
3. Optionally regenerate the dictionary
4. Deploy the typelist

Step 1: Create the typelist file



- In Project View, select `.../config/Extensions/Typelists`
 - Context menu → New → Typelist
- Typelist dialog
 - Naming convention is for custom typelists to end in `_Ext`
 - Example: `InteractionReason_Ext`

Step 2: Define typecodes



- Add new typecodes with toolbar or context menu and define typecode attributes in the Name Value pane
 - Code is internal reference, must be unique within typelist, <=50 characters and contain alphanumeric values
 - Name is the default for when displayed in the user interface and Desc for the data dictionary
 - Priority is the sort order

Step 3: Optionally regenerate dictionary

- gwXX regen-dictionary
- Process builds entire entity model including base and custom typelists
- Identifies errors in the data model beyond Typelist Editor schema validation

```
C:\Guidewire\TrainingApp\bin>gwta regen-dictionary
regen-entity-model-xml:
=====
= Running main class:
  com.guidewire.tools.dictionary.data.EntityModelXmlTool
  [java] --- Guidewire Entity Model In Xml ---
...
ERROR Errors found in DoctorSpecialtyType
ERROR TypelistCategoriesValidator - Typecode
  DoctorSpecialtyType.Critical Care Medicine refers to a
  non-existent category
```

Step 4: Deploy the typelist

Restart Server

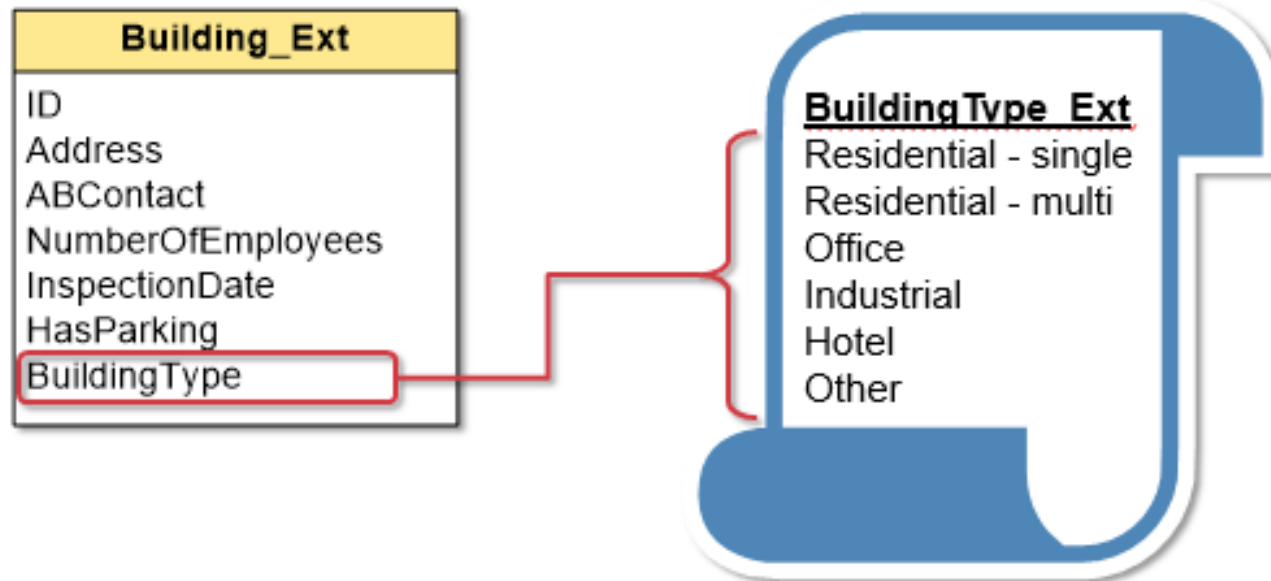
- Typelist



- bin command window
 - **gwXX dev-stop**
 - **gwXX dev-start**

Defining Typekey fields

- A **typekey field** is an entity defined field associated with a specific typelist
- Referenced typelist contains typecodes whose values are the only possible value for the typekey field



Entity configuration: typekey

Building_Ext.eti x

+ fulldescription Show all

Element	Primary Value	Secondary Value
entity	Building_Ext	Information about a ...
foreignkey	Address	Address
foreignkey	ABContact	ABContact
column	NumberOfEmployees	integer
column	InspectionDate	datetime
column	HasParking	bit
typekey	BuildingType	BuildingType_Ext

Name	Value
name	BuildingType
typelist	BuildingType_Ext
nullok	<input checked="" type="checkbox"/> true
desc	Building type
columnName	
createhistogram	<input type="checkbox"/> false
default	
typefilter	

Entity Xml

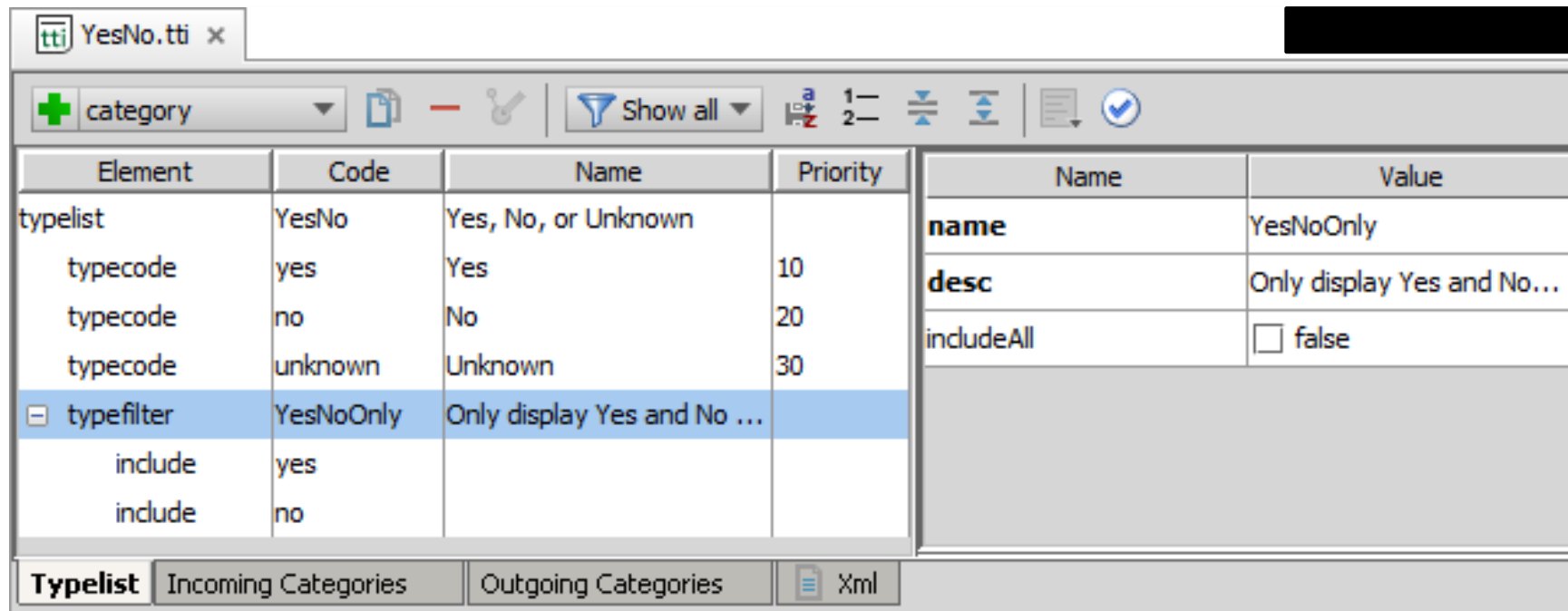
Task – Create Typelist

- Create a typelist `Occupation_Ext.tti` and add the following codes:
 - Doctor
 - Nurse
 - Lawyer
 - Police
- Add a typekey in `HouseDetails_Ext` to refer this typelist

Type Filter

Typelist filters

- A **typelist filter** defines a subset of typecodes in the typelist
- Configure filters for typelists to filter available typecodes
- Example: YesNoOnly typefilter



The screenshot shows a configuration window titled "YesNo.tti". It features a toolbar with a "category" dropdown, a "Show all" button, and various icons. The main area contains two tables. The left table lists elements of the typelist, including typecodes and a typefilter. The right table shows the configuration for the selected "typefilter".

Element	Code	Name	Priority
typelist	YesNo	Yes, No, or Unknown	
typecode	yes	Yes	10
typecode	no	No	20
typecode	unknown	Unknown	30
<input checked="" type="checkbox"/> typefilter	YesNoOnly	Only display Yes and No ...	
include	yes		
include	no		

Name	Value
name	YesNoOnly
desc	Only display Yes and No...
includeAll	<input type="checkbox"/> false

At the bottom, there are tabs for "Typelist", "Incoming Categories", "Outgoing Categories", and "Xml".

Referencing a typelist filter

- For the typekey element in the entity file, specify the typefilter attribute
- Only values in the typefilter are specified (included or excluded)

The screenshot displays two overlapping windows from a software application. The background window, titled 'ABPolicyPerson.etx', shows a table of elements. The foreground window, titled 'YesNo.tti', shows a typelist configuration.

ABPolicyPerson.etx Table:

Element	Primary Value	Secondary Value
subtype (extension)	ABPolicyPerson	Contact type repres...
column	CollateralRequired	bit
column	CollateralAmount	integer
typekey	PersonCollateralVerified	YesNo
+ column	HeightInMeters	decimal

YesNo.tti Table:

Element	Code
typelist	YesNo
typecode	yes
typecode	no
typecode	unknown
typefilter	YesNoOnly
include	yes
include	no

A red box highlights the 'typekey' row in the 'ABPolicyPerson.etx' table. Another red box highlights the 'typefilter' row in the 'YesNo.tti' table. A red arrow points from the 'typefilter' row in the foreground window to the 'typefilter' row in the background window, indicating the reference.

Task – Create Type Filter

- Create typefilter to display occupation related to Hospital
- Refer the typefilter in HouseDetails_Ext

Data Dictionary

Data Dictionary

- Documents the entities and typelists in the application
- <install directory>\build\dictionary\data\index.html
Eg : <C:\PolicyCenter\build\dictionary\data\index.html>
- To regenerate dictionary, from bin directory, execute:
gwXX regen-dictionary

Summary

- Data Modal
 - Entity – eti , eix , etx
 - Entity Fields – define the state or nature of a specific instance of the entity
 - Typelists – tti , tix , ttx
- Extending and Creating Entities
 - Datakey – eg. HouseName
 - Arraykey – eg. RouteDetails
 - Foreignkey – eg. HouseDetails
- Typelist and Typefilters
 - Typekey – eg. Occupation
 - TypeFilter – eg. Hospital
- Data Dictionary - Documents the entities and typelists in the application
 - gwXX regen-dictionary

Thank You