



Subtypes



December 9, 2013

© Guidewire Software, Inc. 2001-2013. All rights reserved.
Do not distribute without permission.

Guidewire training materials contain Guidewire proprietary information that is subject to confidentiality and non-disclosure agreements. You agree to use the information in this manual solely for the purpose of training to implement Guidewire software solutions. You also agree not to disclose the information in this manual to third parties or copy this manual without prior written consent from Guidewire. Guidewire training may be given only by Guidewire employees or certified Guidewire partners under the appropriate agreement with Guidewire.

Lesson objectives

- By the end of this lesson, you should be able to:
 - Describe the role of subtypes and supertypes in the data model
 - Extend an existing subtype entity
 - Create new a subtype entity
 - Create a subtype extension

This lesson uses the notes section for additional explanation and information.
To view the notes in PowerPoint, select View → Normal or View → Notes Page.
When printing notes, select Note Pages and Print hidden slides.

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

2

G U I D E W I R E



Lesson outline

- Subtype basics
- Create a subtype entity extension
- Create a subtype entity

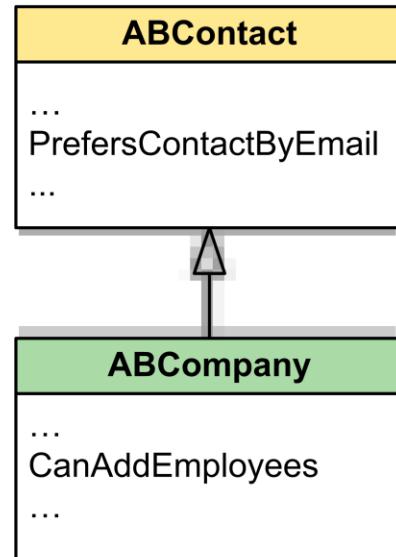
© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

3

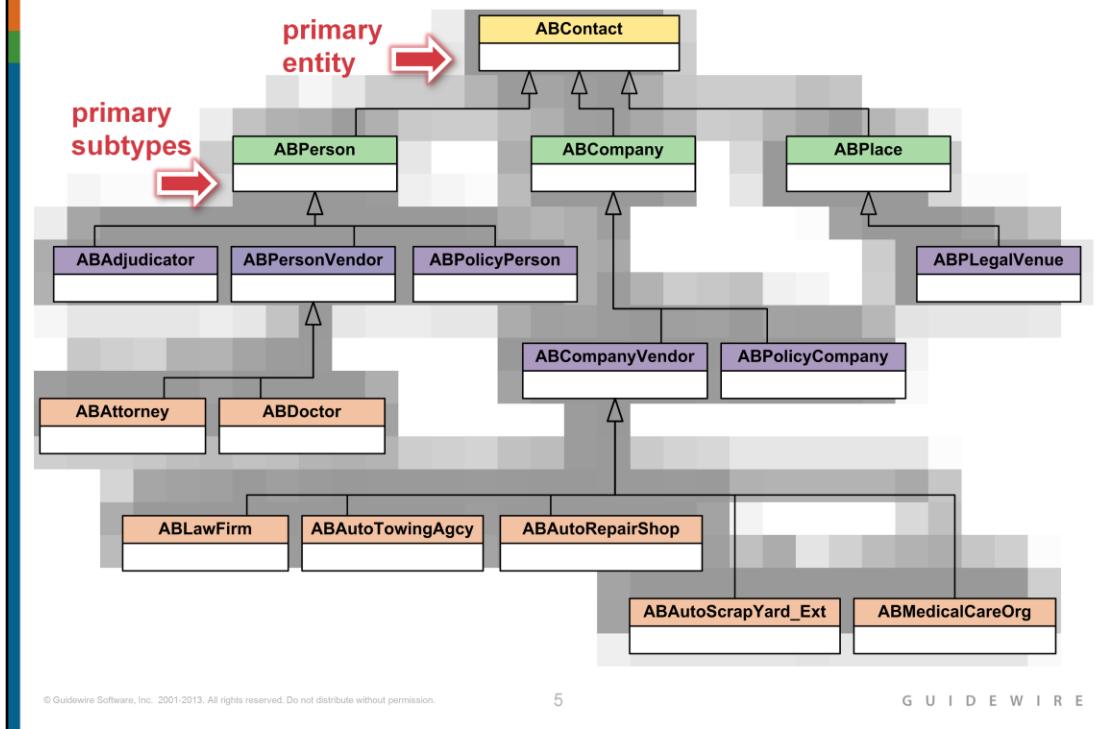
G U I D E W I R E

Subtype entity

- A **subtype** is an entity that is a child to a supertype entity and inherits all fields from the supertype
- Example:
 - ABContact is the supertype
 - ABCompany is the subtype
 - ABCompany inherits PrefersContactByEmail from ABContact
 - ABContact does **NOT** inherit CanAddEmployees



Subtype hierarchy



There is one subtype entity excluded for the diagram because it is not directly relevant to the configuration lab work: ABUserContact. ABUserContact represents a TrainingApp user and defines a user's contact details.

ABContacts are organized into a set of subtypes. The organization of the hierarchy helps to model information about contacts. Information common to all contacts can be established at the ABContact level. It is automatically inherited by all of its subtypes. Information specific to an ABPerson can be defined at the ABPerson level. It is inherited by all of its subtypes, but information on the ABPerson subtype is not available to ABCompany or ABPlace.

In the ABContact hierarchy, an ABPolicyPerson is a person who owns a policy issued by the carrier (such as an individual with a personal auto policy). An ABPolicyCompany is a company which owns a policy issued by the carrier (such as a construction company with a workers' compensation policy).

Subtyping is also used within the three primary applications. For example:

- BillingCenter's Activity, Plan, and Contact entities
- ClaimCenter's Transaction, Incident, and Contact entities
- PolicyCenter's Job and Contact entities

Additional parent and subtype examples

- All applications
 - **Contact:** Person, Company, Place, ...
- ClaimCenter
 - **Incident:** InjuryIncident, TripIncident, PropertyIncident
 - **Transaction:** Payment, Recovery, Reserve, ...
- PolicyCenter
 - **Job:** Submission, Renewal, Cancellation, Reinstatement, ...
 - **PlanDetail:** BillingPlanDetail, PaymentPlanDetail, ...
- BillingCenter
 - **Plan:** BillingPlan, CommissionPlan, DelinquencyPlan, ...
 - **ChargePattern:** ImmediateCharge, ProRataCharge, ...

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

6

GUIDEWIRE

In ClaimCenter:

A transaction is used to denote the movement of money either into a reserve line or from a reserve line to a payment.

An incident reports the details of an incident as it relates to a claim.

In PolicyCenter:

A job is used to manage a policy transaction, which either creates or modifies a policy.

A plan detail is used to manage plans, which typically identify how the insured will be billed for or will pay for the policy. This information is passed to the billing system.

In BillingCenter:

A plan is used to store information about how a billing instruction coming from a policy administration system should be divided into invoices or statements.

A charge pattern is used to divide a charge from the billing instruction (such as a premium for coverages on the policy or a tax mandated by the government) into invoice items.

Entity Editor shows subtypes

- Element reads as subtype or subtype (extension)
 - In XML, it is the <subtype/> or <extension />
- Attribute pane details supertype entity

The screenshot displays two Entity Editors side-by-side. The top Entity Editor is for 'ABAAutoScrapYard_Ext.eti' and the bottom one is for 'ABPersonVendor.etx'. Both editors show tables for attributes and their values.

ABAAutoScrapYard_Ext.eti Entity Editor:

Element	Primary Value	Secondary Value	Name	Value
subtype	ABAAutoScrapYard_Ext	Vendor scrap yard for autos	entity	ABAAutoScrapYard_Ext
column	CanPickUpVehicles	bit	supertype	ABCCompanyVendor
			desc	Vendor scrap yard fo...
			abstract	false

ABPersonVendor.etx Entity Editor:

Element	Primary Value	Secondary Value	Name	Value
subtype (extension)	ABPersonVendor	Contact type repres...	entity	ABPersonVendor
column	SelfEmployee_Ext	bit	supertype	ABPerson
column	FraudConvictions_Ext	integer	desc	Contact type repr...
column	FraudConvictionsDate_Ext	datetime	abstract	false

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

7

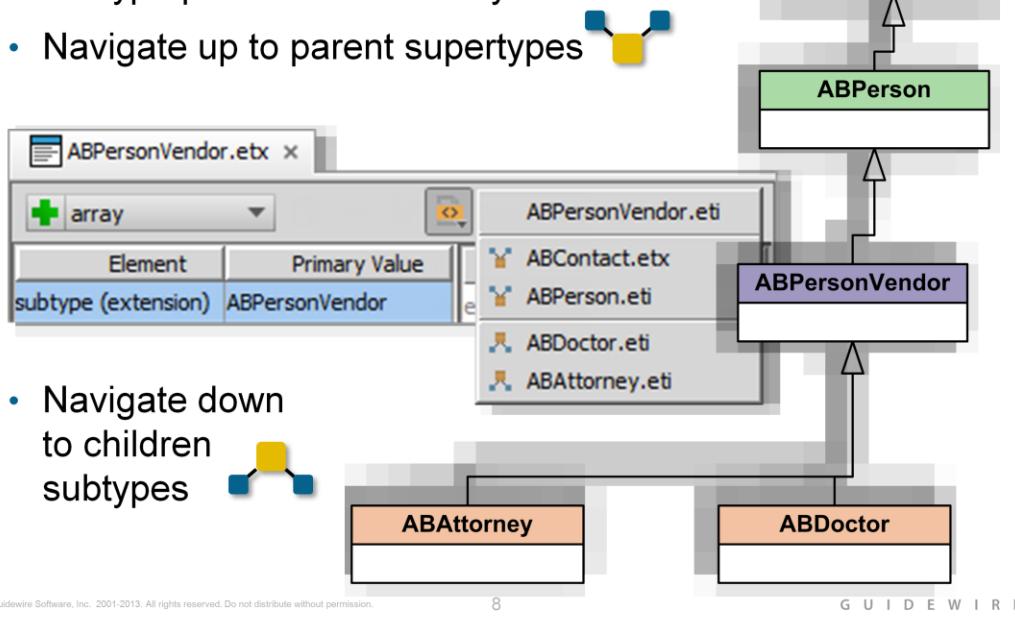
G U I D E W I R E

You can easily identify a subtype in Guidewire Studio using the Entity Editor. The top element will read either subtype or subtype (extension). Not all entities can be extended or subtyped.

If an entity is used to create new supertype entity, then all subtype entities will use the <subtype /> XML element. If the entity is an extension of an existing subtype entity, then the subtype extension uses the <extension /> element. Both elements <subtype /> and <extension /> share the same set of possible sub-elements.

Entity editor toolbar open file navigation

- Open file navigation menu details subtype position in hierarchy
- Navigate up to parent supertypes



In the slide example, ABPersonVendor.etx is a subtype extension of ABPersonVendor.eti. Using the open file navigation menu, you can navigate up to the supertype or down to the child subtype. The open file navigation menu is a quick way to understand how a given subtype or subtype extension relates to a hierarchy of parent supertypes and children subtypes.

Subtypes in the database

- Typelist and entity database tables define supertypes and subtype relationship
 - ABContact has 18 subtypes
- Supertype entity table contains all subtype columns in denormalized form
 - Irrelevant columns for subtype are NULL
 - ID, Name, TaxID from ABContact
 - FirstName, LastName from ABPerson
 - AttorneyLicense from ABAttorney
 - IsFranchise is from ABAutoRepairShop

SELECT ID, TYPECODE FROM ABTL_ABCONTACT ORDER BY ID;	
ID	TYPECODE
1	ABAdjudicator
2	ABAAttorney
3	ABAAutoRepairShop
4	ABAAutoScrapYard_Ext
5	ABAAutoTowingAgcy
6	ABCompany
7	ABContact
8	ABDoctor
9	ABLawFirm
10	ABLegalVenue
11	ABMedicalCareOrg
12	ABPerson
13	ABPlace
14	ABPolicyCompany
15	ABPolicyPerson
16	ABUserContact
17	ABCompanyVendor
18	ABPersonVendor

Select ID, TaxID, FirstName, LastName, AttorneyLicense, Name, IsFranchise, SUBTYPE FROM AB_ABContact

ID	TAXID	FIRSTNAME	LASTNAME	ATTORNEYLICENSE	NAME	ISFRANCHISE	SUBTYPE
67	2096542-3113-2456902	Lily	Watson	J12-13562	null	null	2
68	9983200-5335-0023899	James	Andersen	S20-82325	null	null	2
75	1242577-7777-7752421	null	null	null	Express Auto	null	3
76	3219251-8888-1529123	null	null	null	European Autoworks	null	3

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

9

GUIDEWIRE

ABContact is the supertype for the three primary subtypes : ABPerson, ABCompany, and ABPlace. There are 18 subtypes in total for TrainingApp.

NULL values appear for columns not relevant to a given row's subtype. For example, Lily Watson is an ABAttorney and therefore the Name column is null value. Similarly, Express Auto, an ABAutoRepairShop, shows null values for FirstName and LastName. TaxID is a field defined in the ABContact entity. Both Express Auto and Lily Watson have TaxID encrypted values.

High degrees of subtyping of a single supertype can result in a highly denormalized database table with many columns and large row sizes. For purists seeking high degrees of normalization in database tables, subtyping will appear abhorrent.

In Guidewire applications, subtypes represent a trade-off between data model configuration flexibility and database performance. High degrees of normalization require numerous joins when querying. In many ways, it can be easier to add columns to a table rather than create more table objects in a database that require numerous joins for queries. In the hypothetical case of normalized entities instead of subtypes, to retrieve all results for all contacts would require numerous union queries.

Not all Guidewire applications rely heavily on subtypes, but ContactManager does, and TrainingApp derives from ContactManager. For many configurators, it may make more sense to avoid subtyping in many cases and simply create separate entities that share common typelists and/or arrays.

Subtypes in the data dictionary

- Parent entity

The screenshot shows the Guidewire Data Dictionary interface for the **ABContact** entity. The entity has the following properties:

- Description:** Delegates to [CommonContact](#), [ABLLinkable](#).
- Attributes:** Abstract[?], Editable[?], Exportable[?], Extendable[?], Keyed[?], Loadable[?], Sourceable[?], Supertype[?], Versionable[?]
- Subtypes:** A table listing subtypes with their corresponding codes and names:

Code	Name
ABCCompany	ABCCompany
ABCCompanyVendor	ABCCompanyVendor
ABAutoRepairShop	ABAutoRepairShop
ABAutoTowingAgcy	ABAutoTowingAgcy
ABLawFirm	ABLawFirm
ABMedicalCareOrg	ABMedicalCareOrg
ABPolicyCompany	ABPolicyCompany

Fields:

- ID** key (non-null)[?]: Internally managed primary key.

PrimaryAddress foreign key to [Address](#) (database column: PrimaryAddressID) (exportable)[?] (loadable)[?] (triggers validation)[?] (writable)[?]: Primary address.

ABCompany (subtype of [ABContact](#))

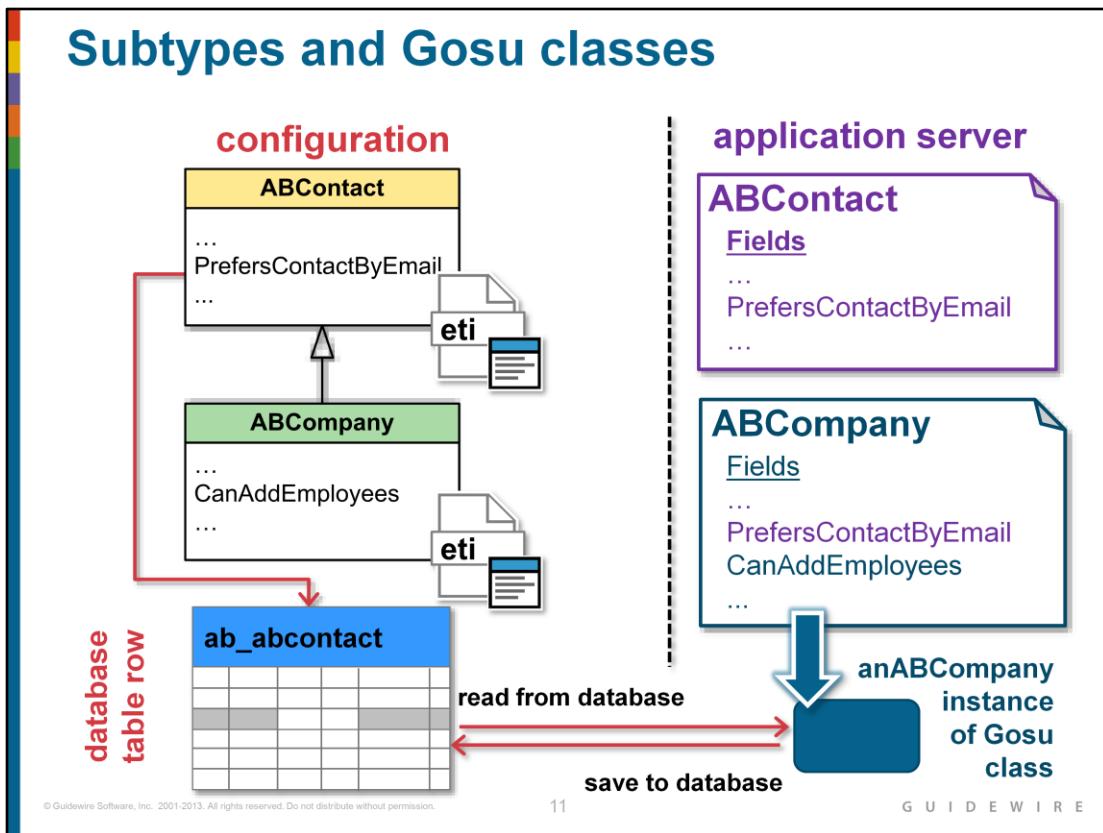
- Description:**
- CanAddEmployees** boolean (default: true) (exportable)[?] (loadable)[?] (writable)[?]: Can employees be added to this company?
- Show user interface details...**
- Case** array key for [ABContact](#) (loadable)[?]: Case.
- Employees** array key for [ABPerson](#) (loadable)[?]: Employees.

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

10

G U I D E W I R E

Subtypes and Gosu classes



In object-oriented design, a class can be extended by a child class or subclass. Every field and method declared at the parent class is added to the child class automatically. In object-oriented design, subclass extensions from a superclass is called inheritance.

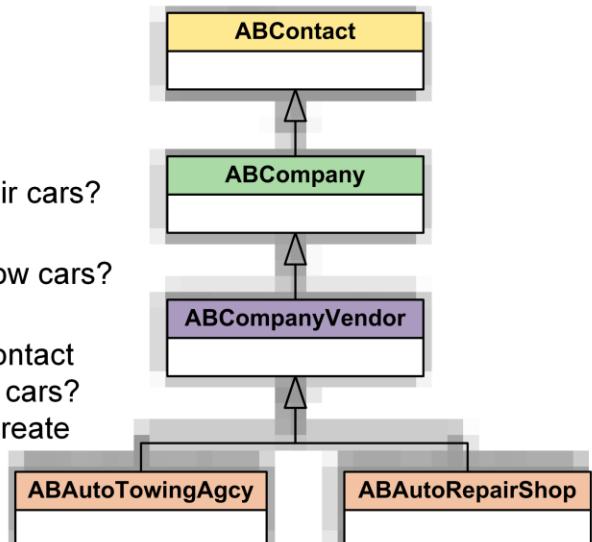
Guidewire subtypes also make use of inheritance. Every subtype has its own internal class. This class inherits all fields and methods of the parent class.

In most cases, a database table corresponds to one data model entity, and therefore one database-backed Gosu class. One exception to this is where subtypes are concerned. If an entity is subtyped, then the database table corresponds to the top-level data model entity and all of its subtype entities. The table then also corresponds to one top-level database-backed Gosu class and all of its subclasses.

In the diagram above, the row in the database table that corresponds to anABCompany is not shown as a continuous strip of gray, but rather as a discontinuous set of gray strips. This is to emphasize the fact that the table for a subtyped entity has columns that do not apply to all subtypes. For example, the **ab_abcontact** table has columns for `FirstName` and `LastName`. These fields only apply to ABContacts that are of type **ABPerson**. For **ABCompanies** (such as **anABCompany**), those columns will have null values.

Subtype limitations

- New subtypes inherit from supertype
 - All elements and attribute defaults
 - Behavior of its supertype
 - Only single inheritance
- Unable to inherit from multiple supertypes
 - Can a towing agency repair cars?
No.
 - Can an auto repair shop tow cars?
No.
 - Can you create a single contact that both repairs and tows cars?
Yes, but you will need to create a new subtype with duplicate fields!



© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

12

G U I D E W I R E

Consider carefully before manipulating your contact hierarchy. A new subtype inherits the attributes and matching behavior of its supertype. Create a new subtype only if you need to treat one set of entities differently from another. As much as possible, limit the number of subtypes for representing an entity.

The main reason to limit creation of new subtypes is that the more you specialize the subtype hierarchy, the more restrictive and the less flexible your model becomes. For example, there are subtypes for AutoRepairShop and AutoTowingAgcy. If you work with an auto repair shop that also does towing, you cannot create a single contact that does both. However, you can add a service for Towing to an auto repair shop contact.

You cannot create a subtype that inherits the fields of both entities. To create the subtype you want, you can select one entity or the other and subtype and then add the fields that are missing from the other entity because of single inheritance.

Additionally, each time you create a new subtype, you must modify PCF files to support both creating the subtype and searching for it. You might also need to make supporting modifications to the screens that reference subtype in terms of modal PCF support.



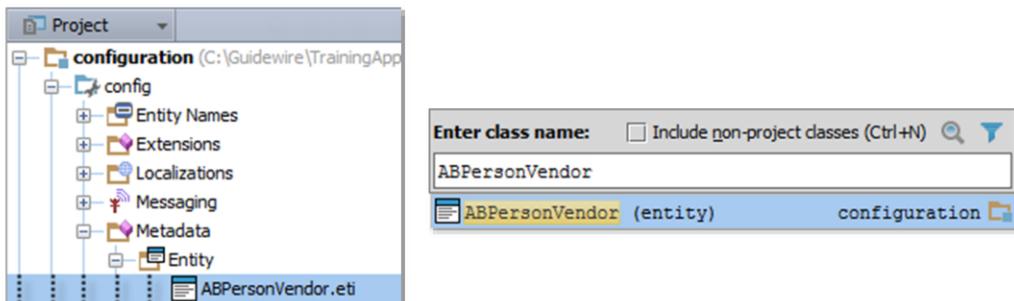
Lesson outline

- Subtype basics
- Create a subtype entity extension
- Create a subtype entity

Steps to create a subtype extension

1. Navigate to the subtype entity
2. Create an entity extension file
3. Add elements (and subelements) and specify attribute values
4. Optionally regenerate the dictionary
5. Deploy the subtype extension entity

Step 1: Navigate to the subtype entity (1)



- Navigate to an entity in ...**Metadata\\Entity**
 - Project View or using CTRL+N
- Studio will open an existing extension first
 - If an existing extension already exists, STOP
 - Do NOT create a new extension; edit existing extension instead!
- Verify that you have selected an entity file (ETI)
 - Not EIX or ETX

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

15

G U I D E W I R E

Step 1: Navigate to the subtype entity (2)

- View in Entity Editor or Xml Viewer
- Entity (ETI) file must be a subtype entity
 - <subtype /> element
 - supertype = entity attribute name value
 - final = false attribute name value

```
<subtype ... entity="ABPersonVendor" final="false" supertype="ABPerson">
```

Element	Primary Value	Secondary Value
subtype	ABPersonVendor	Contact type repr...

Name	Value
entity	ABPersonVendor
supertype	ABPerson
desc	Contact type repre...
final	<input type="checkbox"/> false

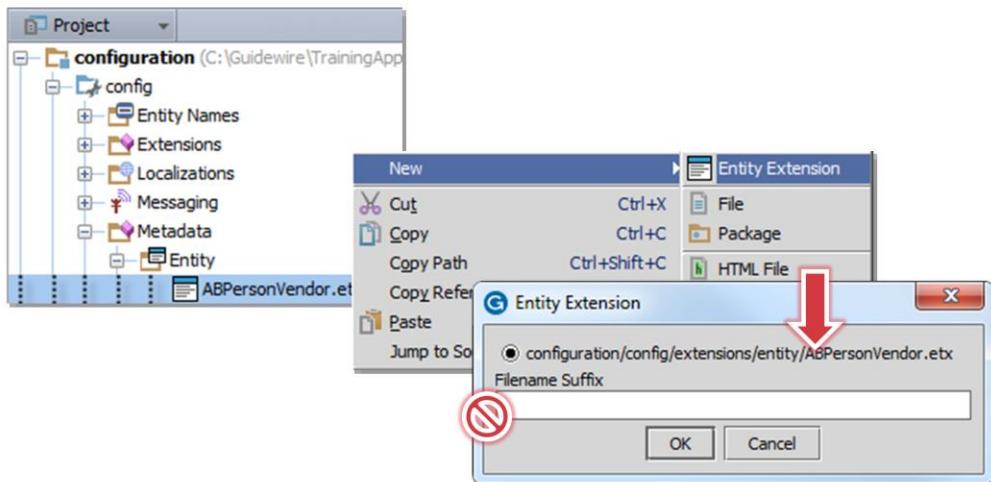
© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

16

G U I D E W I R E

Step 2: Create an entity extension file (1)

- Project View → Context menu → New → Entity Extension
- Do **NOT** enter filename suffix; Click OK

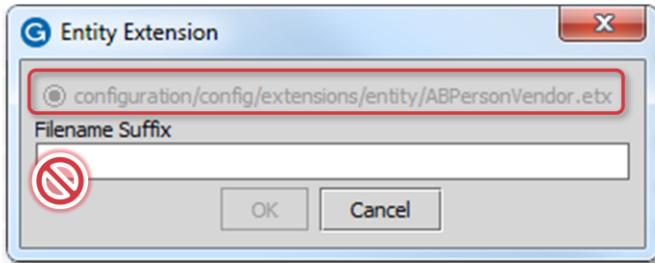


© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

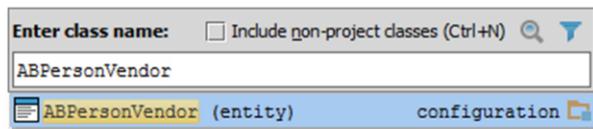
17

G U I D E W I R E

Step 2: Create an entity extension file (2)



- Unable to click OK? Click Cancel
 - Grayed out path shows that an entity extension (ETX) file already exists!
- Navigate to the entity (CTRL+N)
 - Studio automatically opens the extension file first!
 - Edit the extension in the Entity Editor



© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

18

G U I D E W I R E

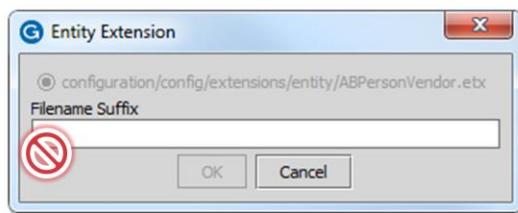
If an extension already exists, simply edit the extension rather than creating another extension using a suffix.

Step 2: Create an entity extension file (3)

- For an entity (ETI) in ...**Extensions\\Entity**
 - NOT possible to create entity extension (ETX)
 - Edit the ETI file directly in the Entity Editor

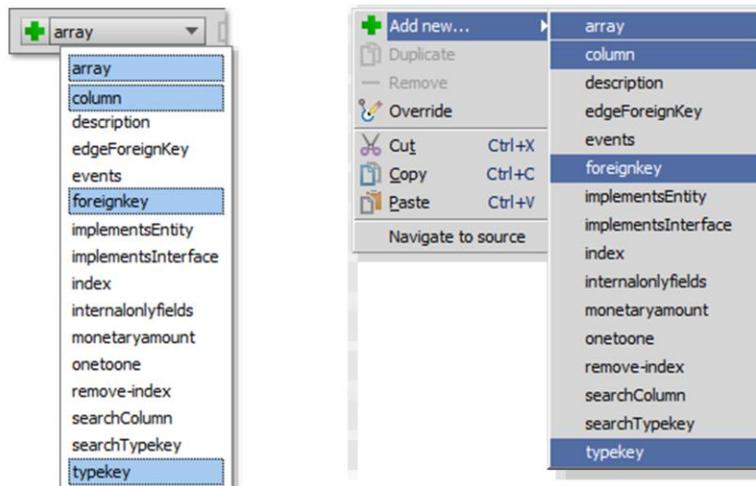


- For an entity extension (ETX) in ...**Extensions\\Entity**
 - Do **NOT** create an extension for an extension
 - Click Cancel
 - Edit the ETX file directly in the Entity Editor



Step 3: Add elements and specify attributes

- Toolbar
 - Select option in dropdown
 - Click + to add more of same
- Context menu
 - Add new...
 - Select option in menu



© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

20

G U I D E W I R E

The four primary elements listed above are common to entity extension (<extension>), entity declaration (<entity>), subtype extension (<extension>), and subtype declaration (<subtype>).

Common elements to add

- Common elements for:
 - Entity extension, entity, subtype, subtype extension
- <array .../>
 - Define array entity and field
- <column .../>
 - Define data field with defined data type
 - Bit, datetime, integer, varchar
- <foreignkey .../>
 - Define foreign key field and entity
- <typekey .../>
 - Define typekey and related typelist
- **Create entity extension field names ending with _Ext**

The four primary elements listed above are common to entity extension (<extension>), entity declaration (<entity>), subtype extension (<extension>), and subtype declaration (<subtype>).

Validate the schema



Click Validate in the toolbar

- Red highlight indicates schema violation warning
- Schema violations explained in pane below editor

The screenshot shows the ABPersonVendor.etx entity editor. The main table lists entity properties with columns for Element, Primary Value, Secondary Value, Name, and Value. The 'Secondary Value' column for the 'subtype (extension)' row contains a red highlight. Below the table, tabs for Entity and XML are visible, with Entity selected. A message bar at the bottom displays a red error message: "NoTwoColumnsWithTheSameName - Duplicate property 'SelfEmployeed_Ext' found for entity 'ABPersonVendor', (...".

Element	Primary Value	Secondary Value	Name	Value
subtype (extension)	ABPersonVendor	Contact type repre...	entity	ABPersonVendor
column	SelfEmployeed_Ext	bit	supertype	ABPerson
column	FraudConvictions_Ext	integer	desc	Contact type repre...
column	FraudConvictionsDate...	datetime	abstract	false
column	SelfEmployeed_Ext	bit	displayName	Vendor (Person)

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

22

G U I D E W I R E

You can also optionally regenerate the data dictionary to check for problems in the data model.

Step 4: 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 ABPersonVendor  
ERROR NoTwoColumnsWithTheSameName - Duplicate property  
  "SelfEmployeed_Ext" found for entity "ABPersonVendor"
```

You can also regenerate the data dictionary to add the new entity to the data dictionary and to check for problems in the data model. Regenerating the data dictionary is not required, but doing so can identify flawed XML in the data model that go beyond schema validation such as certain types of referential integrity.

Step 5: Deploy the subtype entity extension

Restart Server

- Entity Extension

- bin command window
 - **gwXX dev-stop**
 - **gwXX dev-start**
- Or, Guidewire Studio
 - Run → Stop
 - Run 'Server' or Debug 'Server'
- During start-up
 - If autoupgrade=true in database-config.xml, then Guidewire attempts to upgrade the database according to the changes in the data model



© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

24

G U I D E W I R E



Lesson outline

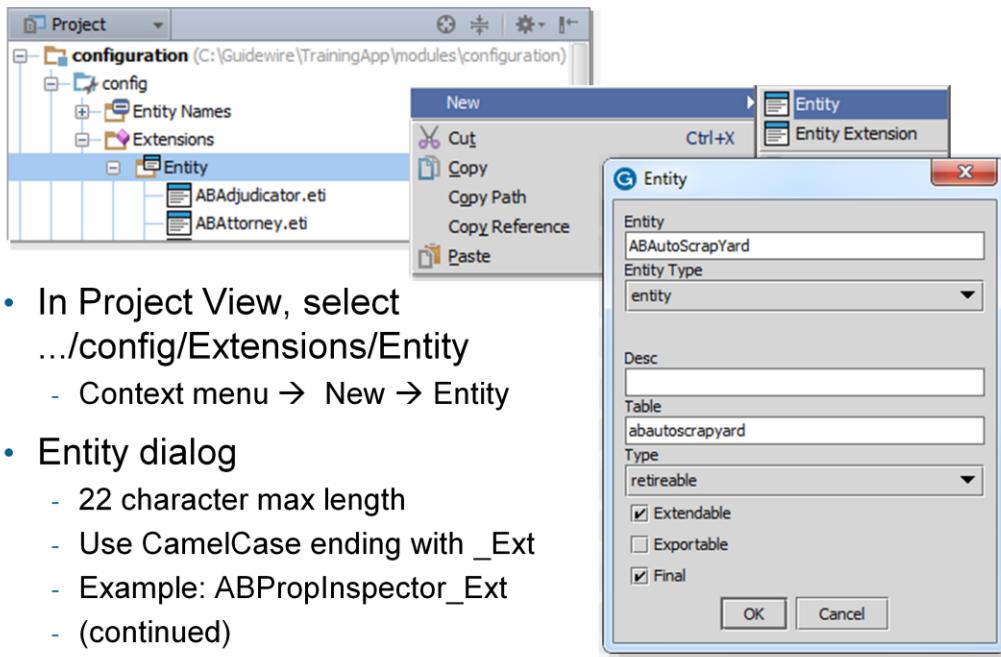
- Subtype basics
- Create a subtype entity extension
- Create a subtype entity

Steps to create a new subtype

1. Create the subtype entity file
2. Add elements and specify attributes
3. Optionally regenerate the dictionary
4. Deploy the subtype entity

Adding a new subtype is similar to the process of adding an entity. However, it requires details including the subtype specification, and the supertype.

Step 1: Create the subtype entity (1)



- In Project View, select .../config/Extensions/Entity
 - Context menu → New → Entity
- Entity dialog
 - 22 character max length
 - Use CamelCase ending with _Ext
 - Example: ABPropInspector_Ext
 - (continued)

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

27

G U I D E W I R E

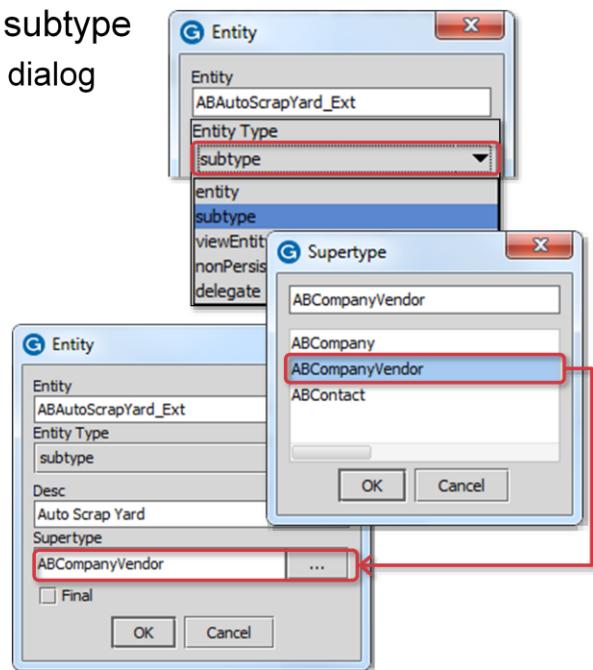
Guidewire recommends the entity name should be in CamelCase. CamelCase is a style of capitalization in which multiple words are joined together with the first character of each word capitalized. Examples include ClaimCenter and JavaScript. CamelCase is the naming convention recommendation for entities. Guidewire recommends ending the name of an Entity with _Ext to distinguish the entity from base application entities.

For every subtyped entity, Guidewire applications create a virtual typelist accessible to the application as well as physical database table that stores the subtype values. The typelists consist of a set of typecodes, one typecode for each subtype. The "code" of each typecode is the subtype's name.

The Guidewire application server (not Studio, unfortunately) limits typecode code names to 22 characters. Therefore, the name of any given subtype cannot exceed 22 characters. If you do create a subtype with 23 or more characters, any attempt to restart the server or regenerate the data dictionary will fail.

Step 1: Create the subtype entity (2)

- For Entity Type, select subtype
 - Fields change in Entity dialog
- Select supertype
 - In Supertype field, click ellipse (...) to open the Supertype dialog
 - In dialog, select the parent entity



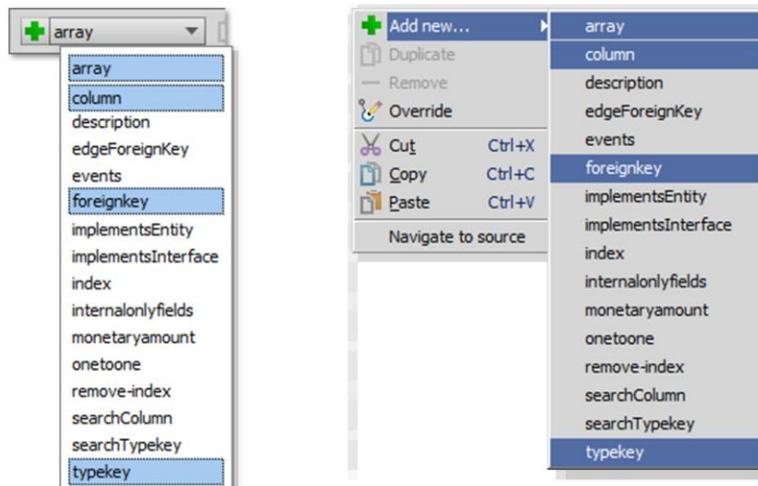
© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

28

G U I D E W I R E

Step 2: Add elements and specify attributes

- Toolbar
 - Select option in dropdown
 - Click + to add more of same
- Context menu
 - Add new...
 - Select option in menu



© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

29

G U I D E W I R E

The four primary elements listed above are common to entity extension (<extension>), entity declaration (<entity>), subtype extension (<extension>), and subtype declaration (<subtype>).

Common elements to add

- Common elements for:
 - Entity extension, entity, subtype, subtype extension
- <array .../>
 - Define array entity and field
- <column .../>
 - Define data field with defined data type
 - Bit, datetime, integer, varchar
- <foreignkey .../>
 - Define foreign key field and entity
- <typekey .../>
 - Define typekey and related typelist
- Do **NOT** create entity field names ending with `_Ext`

The four primary elements listed above are common to entity extension (<extension>), entity declaration (<entity>), subtype extension (<extension>), and subtype declaration (<subtype>).

Entity editor: Attribute pane

Name	Value
name	CanPickUpVehicles
type	bit
nullok	<input checked="" type="checkbox"/> true
desc	Can this yard tow non-operatio...
autoincrement	
columnName	
createhistogram	<input type="checkbox"/> false
default	2
deprecated	<input type="checkbox"/> false
exportable	<input checked="" type="checkbox"/> true
getterScriptability	all
ignoreforevents	<input type="checkbox"/> false
loadable	<input checked="" type="checkbox"/> true
overwrittenInStagingTable	<input type="checkbox"/> false
scalable	<input type="checkbox"/> false
setterScriptability	all
soapnullok	<input type="checkbox"/> false
supportsLinguisticSearch	<input type="checkbox"/> false

- Name Value columns
 - For selected element, define attributes
- Schema aware values
 - Boolean controls
 - Dropdown lists
- Attribute styling
 - Bold for required
 - Black for editable
 - Grayed-out for non-editable
 - Overridden, Inherited, Internal, Default
- Nullok defaults to false!
 - Set to true in most cases

Validate the schema



Click Validate in the toolbar

- Red highlight indicates schema violation warning
- Schema violations explained in pane below editor

Element	Primary Value	Second...	Name	Value
subtype	ABAutoScrapYard_Ext		name	CanPickUpVehicles
column	CanPickUpVehicles	bit	type	bit
column	CanPickUpVehicles	bit	nullok	<input checked="" type="checkbox"/> true
			desc	Can this yard tow non-o...
			autoincrement	
			columnName	

© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

32

G U I D E W I R E

You can also optionally regenerate the data dictionary to check for problems in the data model.

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 ABAutoScrapYard_Ext  
ERROR NoTwoColumnsWithTheSameName - Duplicate property  
  "CanPickUpVehicles" found for entity "ABAutoScrapYard_Ext"
```

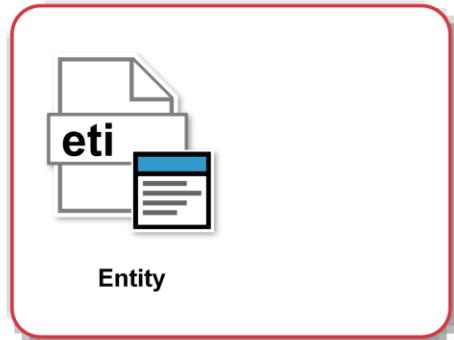
You can also regenerate the data dictionary to add the new entity to the data dictionary and to check for problems in the data model. Regenerating the data dictionary is not required, but doing so can identify flawed XML in the data model that go beyond schema validation such as certain types of referential integrity.

Step 4: Deploy the subtype entity

Restart Server

- Entity

- bin command window
 - **gwXX dev-stop**
 - **gwXX dev-start**
- Or, Guidewire Studio
 - Run → Stop
 - Run 'Server' or Debug 'Server'
- During start-up
 - If autoupgrade=true in database-config.xml, then Guidewire attempts to upgrade the database according to the changes in the data model



© Guidewire Software, Inc. 2001-2013. All rights reserved. Do not distribute without permission.

34

G U I D E W I R E

In Studio, the newly created entity is available to reference immediately. However, the entity is not available to the application server until the server is restarted and the database upgraded.

The automatic database upgrade process occurs only if the database autoupgrade=true attribute is defined in the database-config.xml.

TrainingApp uses an H2 database. Here is the XML definition found in database-config.xml:

```
<database name="TrainingAppDatabase" dbtype="h2" autoupgrade="true">
  <dbcp-connection-pool
    jdbc-url="jdbc:h2:file:/Guidewire/TrainingApp/db/ta;auto_server=true"/>
</database>
```

For SQL Server and Oracle, this option is set to false by default.

Lesson objectives review

- You should now be able to:
 - Describe the role of subtypes and supertypes in the data model
 - Extend an existing subtype entity
 - Create new a subtype entity
 - Create a subtype extension

Review questions

1. What is a subtype entity? What is the XML element for a subtype entity?
2. What is a subtype entity extension? What is the XML element for a subtype entity extension?
3. Describe at least one limitation of relying heavily on subtype hierarchies.
4. When you create a new subtype, why are you NOT able specify a table name?
5. If creating a subtype extension and the path to the extension is grayed out and you are unable to click OK, what should you do?

Answers

- 1) A subtype is an entity that is a child to a parent entity and inherits all fields from the parent. <subtype /> is the XML element for a subtype entity.
- 2) A subtype extension is an entity extension of a subtype entity. <extension /> is the XML element for a subtype extension entity.
- 3) One major limitation of a subtype hierarchy is that a subtype entity can only have one parent entity and multiple inheritance (two or more direct parents) is not supported. In addition, the subtype always inherits the parents elements, meaning the subtype contains the parent fields. For a deep hierarchy (many levels), this means the lowest child will inherit all the fields of the parents. Deep subtype hierarchies can result in very wide tables that are highly denormalized resulting in poor application performance.
- 4) You cannot specify a table name because the new subtype data is stored in the same table as the supertype.
- 5) You should click Cancel and simply edit the existing subtype entity.

Notices

Copyright © 2001-2013 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire ExampleCenter, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. Guidewire products are protected by one or more United States patents.

This material is Guidewire proprietary and confidential. The contents of this material, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This file and the contents herein are the property of Guidewire Software, Inc. Use of this course material is restricted to students officially registered in this specific Guidewire-instructed course, or for other use expressly authorized by Guidewire. Replication or distribution of this course material in electronic, paper, or other format is prohibited without express permission from Guidewire.