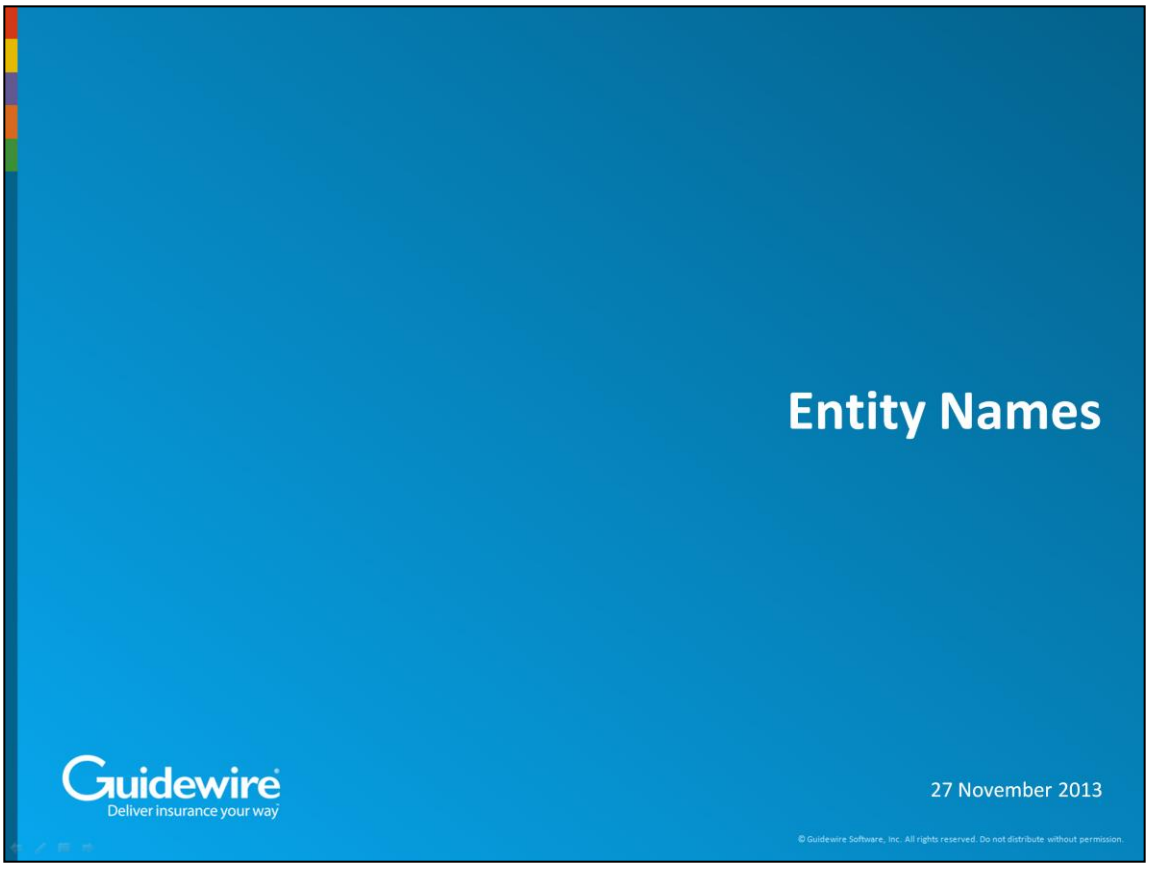


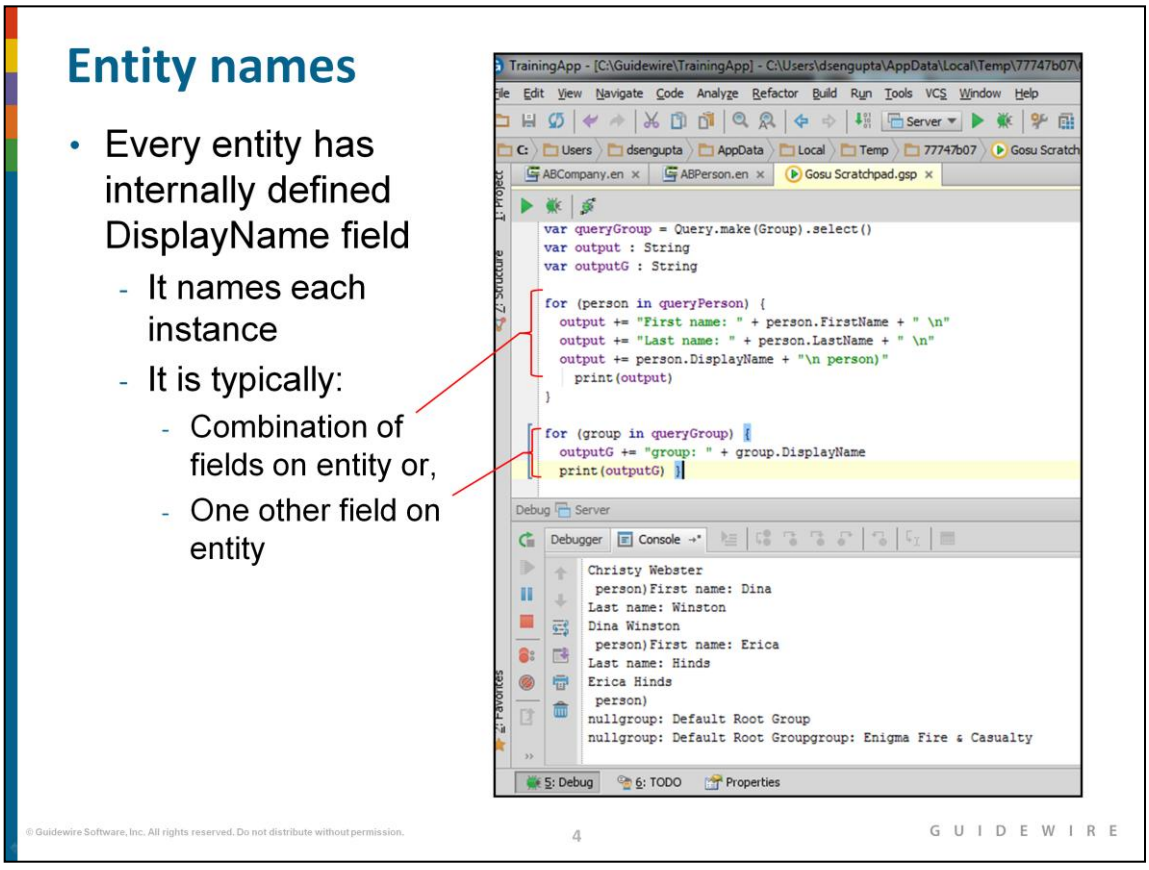
Lesson objectives

- By the end of this lesson, you should be able to:
  - Describe the role of entity names
  - Create and modify entity names

This lesson uses the notes section for additional explanation and information.
To view the notes in PowerPoint, choose View→Normal or View→Notes Page.
If you choose to print the notes for the lesson, be sure to select "Print hidden slides."

Lesson outline

- Entity name basics
- Configuring entity names

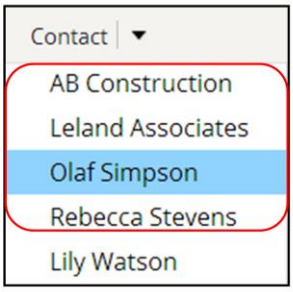


In the example above, the code in the Gosu Scratchpad demonstrates that:

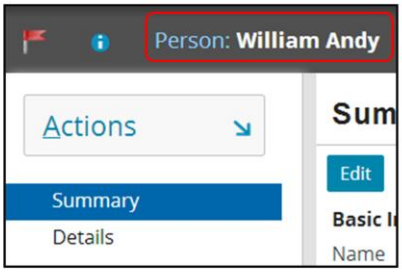
- The display name for a group is simply the value of the group's Name field.
- The display name for an ABPerson is a concatenation of the first name plus a blank space plus the last name. If the person's name also has a suffix, such as "Jr.", that is also concatenated to the value.

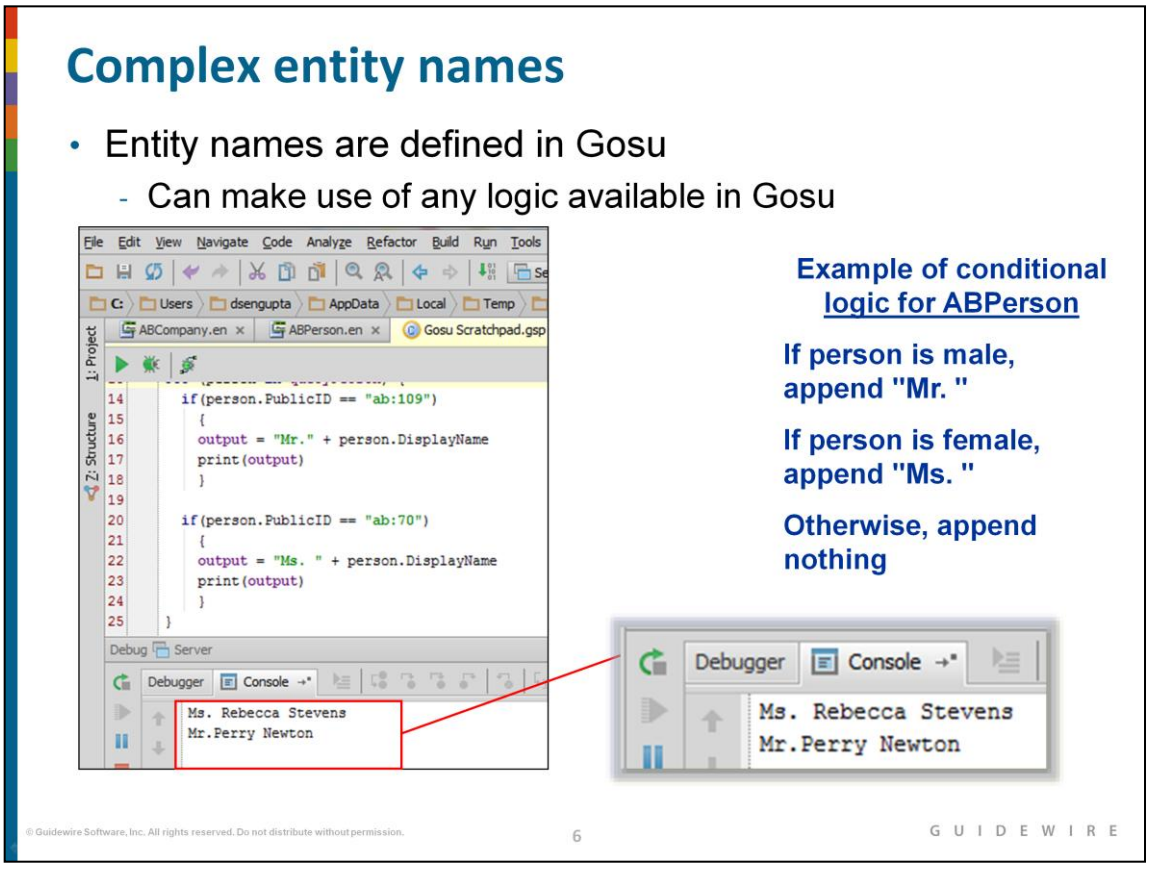
Where entity names are used

- Whenever object as a whole is displayed
  - For example, dropdown that lists ABContacts



- Whenever object's DisplayName is explicitly referenced
  - For example, info bar widget with value property set to ABContact.DisplayName





Because display names are defined in Gosu, display names can use the following aspects of Gosu:

- Conditional logic, such as a display name for ABPerson that appends the middle name field only if it is not null or an activity display name that uses "Open: " plus the subject for open activities and "Closed: " plus the subject for closed activities.
- String manipulation, such as a display name for order numbers that converts the value to all capital letters, or a display name for credit card numbers that uses just the last four digits.
- Calculations, such as a display name for ABCompany that appends the number of employees at the end of the name.

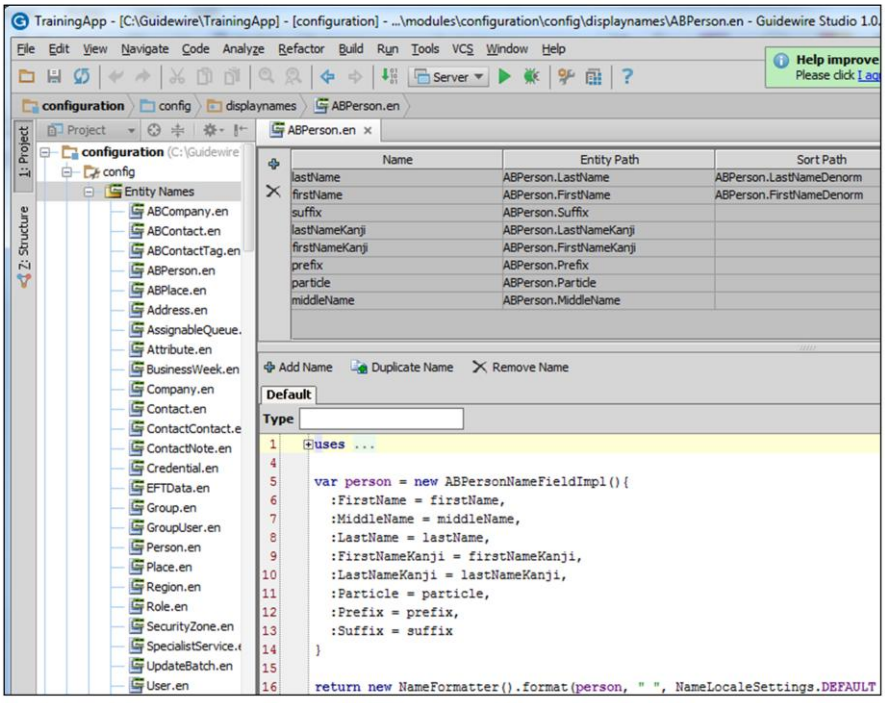
What can be configured?

- Developers can:
  - Create entity names for new entities
  - Modify entity names for existing entities
  - Reference an object's display name in:
    - PCF files
    - Gosu code

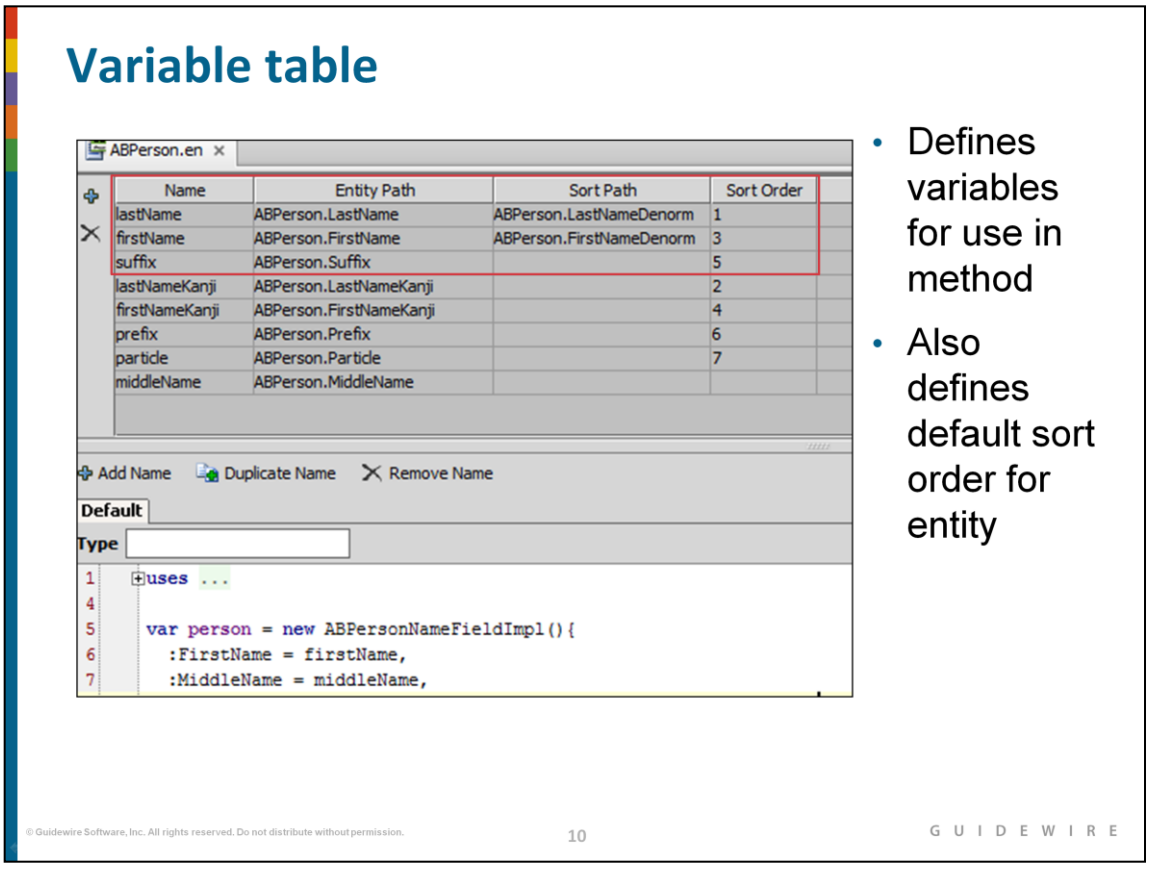
Lesson outline

- Entity name basics
- Configuring entity names

Entity name editor



The entity name editor is a component of Studio used to configure how the display name for a given entity is generated.



The first two columns in the variable table are used for generating the entity name. They are:

- Name - The symbol name, which is used in the return method
- Entity path - The path from the entity to the field whose value should be used by the symbol. Typically, these are fields directly on the entity.

The next two columns in the variable table define the default sort order of the entity. They are:

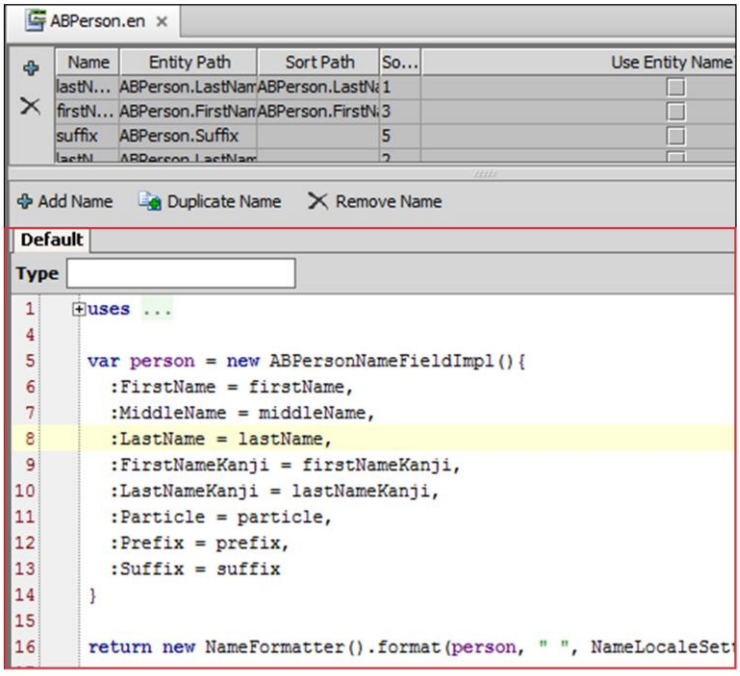
- Sort path - Defines the value to be used in the sort, defaults to the entity path.
- Sort order - Defines the order of precedence for the sort

In the slide above, the LastName variable's sort path is "ABPerson.LastNameDenorm" and the FirstName variable's sort path is "ABPerson.FirstNameDenorm". Therefore, the default sort order is to first sort entities based on the LastNameDenorm field (which is used first because it has a sort order of 1), and then, for any entities with the same LastNameDenorm value, to sort those entities based on the FirstNameDenorm field.

The last column in the variable table, "Use Entity Name?", influences how the given entity name works with foreign keys.

- If the "Entity Path" column designates a field that is a foreign key, then the "Use Entity Name?" column should be set to true. This tells the Guidewire application that it should not load the object that the foreign key points to, but rather it should simply calculate the entity name for the foreign key object and return that value as a string. If the "Entity Path" column designates a field that is a foreign key and the "Use Entity Name?" column is not set to true, then the Guidewire application will unnecessarily load the entire foreign key object. This could negatively impact performance.
- If the "Entity Path" column designates a field that is not a foreign key, then you can either explicitly set the "Use Entity Name?" column to false or leave it blank (in which case it defaults to false). This tells the Guidewire application that there is no foreign key object whose entity name needs to be calculated and returned as a string.

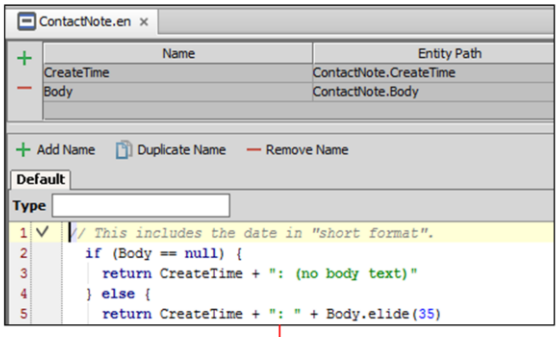
Return value pane

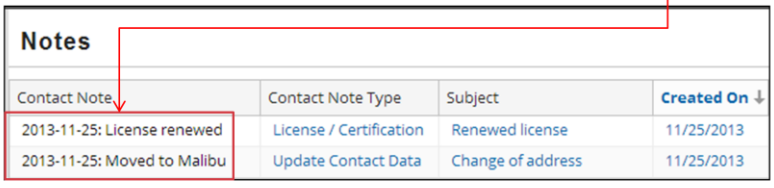


- Determines value to return when object's display name is needed
- If necessary, application converts value to string

Datetime fields in entity names

- When concatenating dates, "short format" is used by default
- If you want "long format", explicitly convert the date to a string

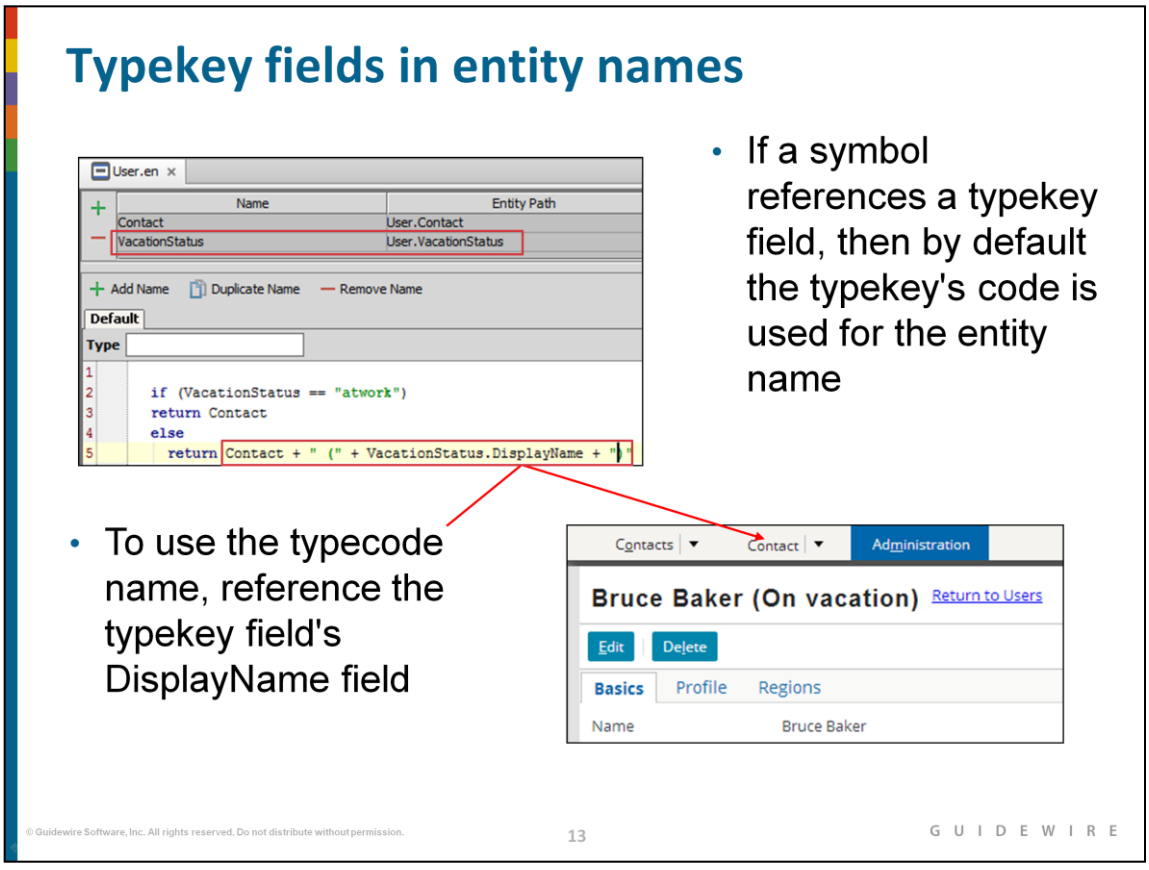




If you return a datetime field in an entity name (without using concatenation), then the datetime's "long format" is used.

If you return a datetime field in an entity name and you use the concatenation operator, then the datetime's "short format" is used.

If you want to concatenate a datetime field in an entity name and get the datetime's long format, then you should convert the datetime field to a string (using the toString() method) before concatenating the value.



If line 4 of the code above was the following...

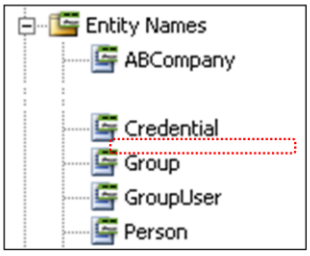
```
return Contact + " (" + VacationStatus + ")"
```

...and Bruce Baker's vacation status was "on vacation", then the label would read as...

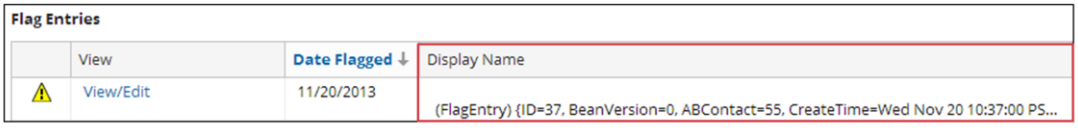
Bruce Baker (onvacation) (Note that the code, "onvacation", is being used, as opposed to the user-friendly name of "On vacation").

You should not create the symbol to reference the field's display name. In the example above, the VacationStatus symbol should not have "User.VacationStatus.DisplayName" as its entity path. Instead, you should reference the display name in the entity name code. There are several reasons for this. Specifying ".DisplayName" in the entity path forces Gosu to do an implicit String coercion, whereas an explicit conversion in the code is considered a better approach. The symbol table is also used to determine the default sort value for the entity. By mapping a symbol to a display name, you can create unpredictable sorting situations.

Entities without entity names



- If an entity (such as FlagEntry) has no entity name and you display object name in UI, then non-user-friendly list of all fields is displayed



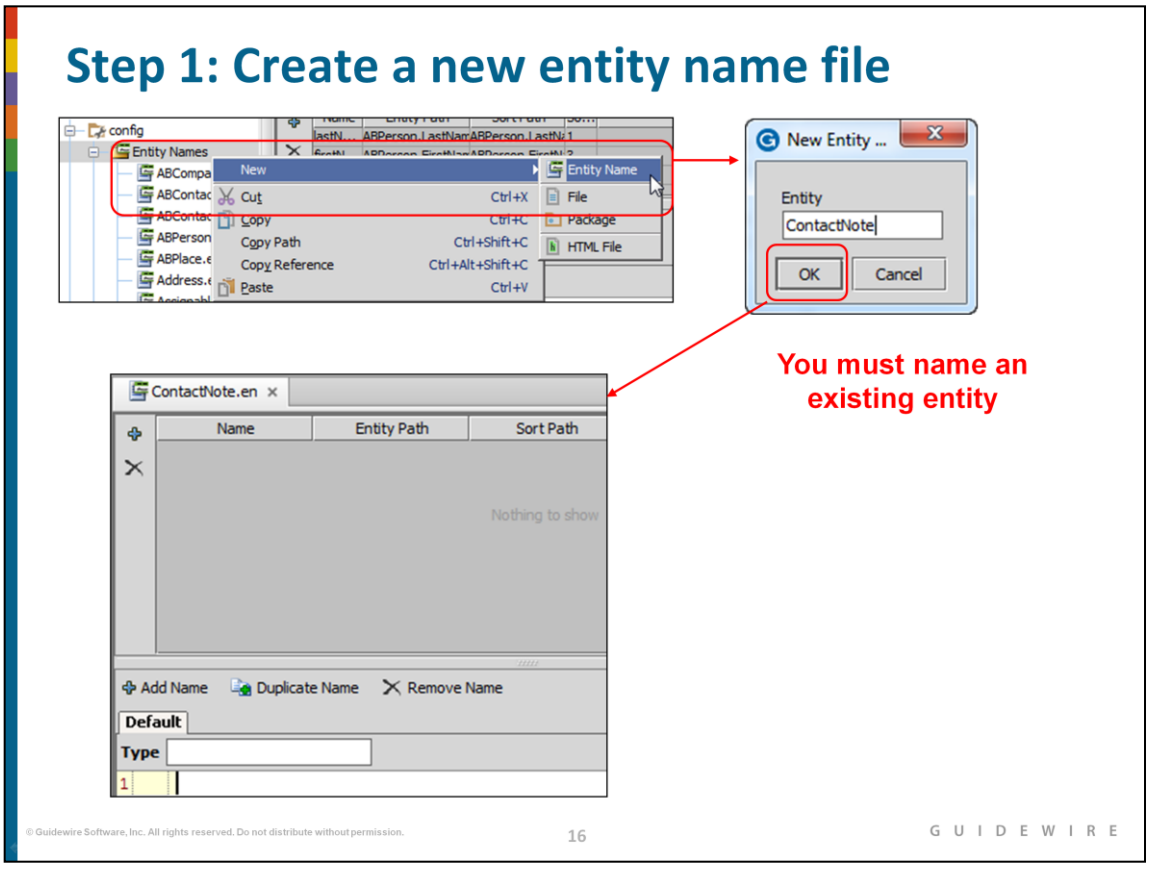
Entities do not inherently have entity names. Typically, the base application provides entity names only for the primary entities (and not for any custom entities).

You should create entity names for:

- Base application entities where the display name is needed but no entity name exists in the base application.
- Custom entities you add to the data model.

Steps to create an entity name

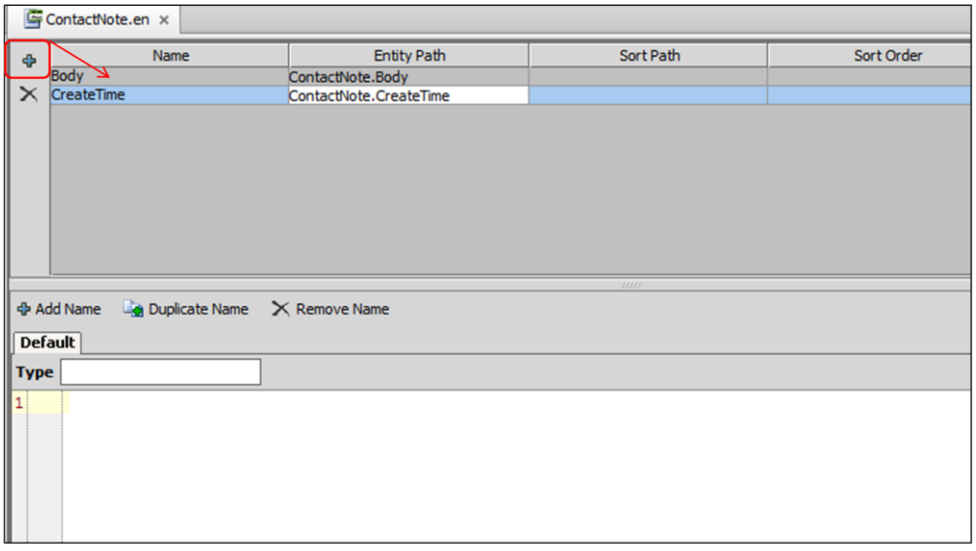
1. Create a new entity name file
2. Define the variables in variable table
3. In the Gosu text editor, select the Default name and specify code for the return value
4. Restart application server



To create a new entity name:

1. Right-click the Entity Names node and select New → Entity Name.
2. In the New Entity Name dialog box, enter the name of an existing entity.
3. Click OK.

Step 2: Define the variables

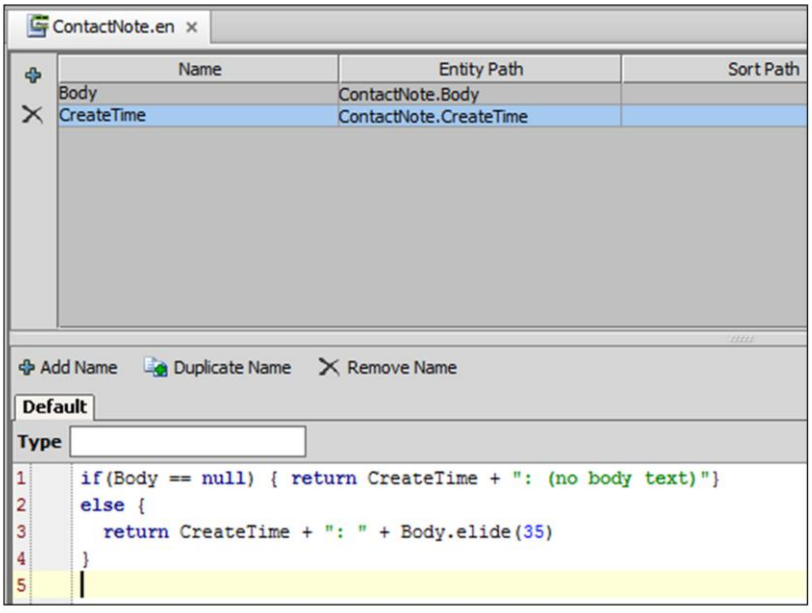


- By convention, Name is usually the same as the entity field name
- Entity path must the path to the field in the entity

GUIDEWIRE

To add a symbol, click the Add button. This creates a blank row in the symbol table. Specify values in the row as needed.
In the Entity Path cell, if you do not enter a valid path for the base entity, the cell displays in red. For example, if enter "ContactNote.Boddy", the cell becomes red because there is no field on ContactNote named "Boddy".

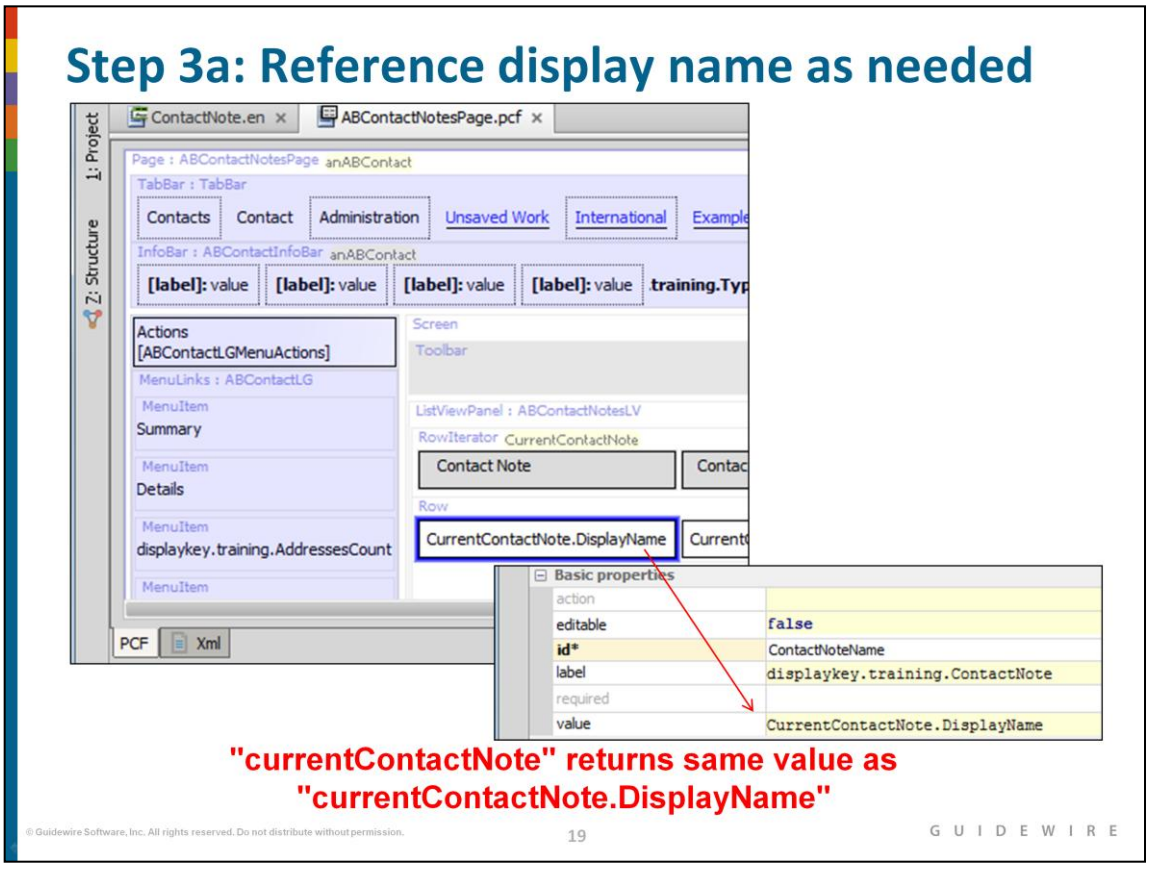
Step 3: Specify code to calculate default value



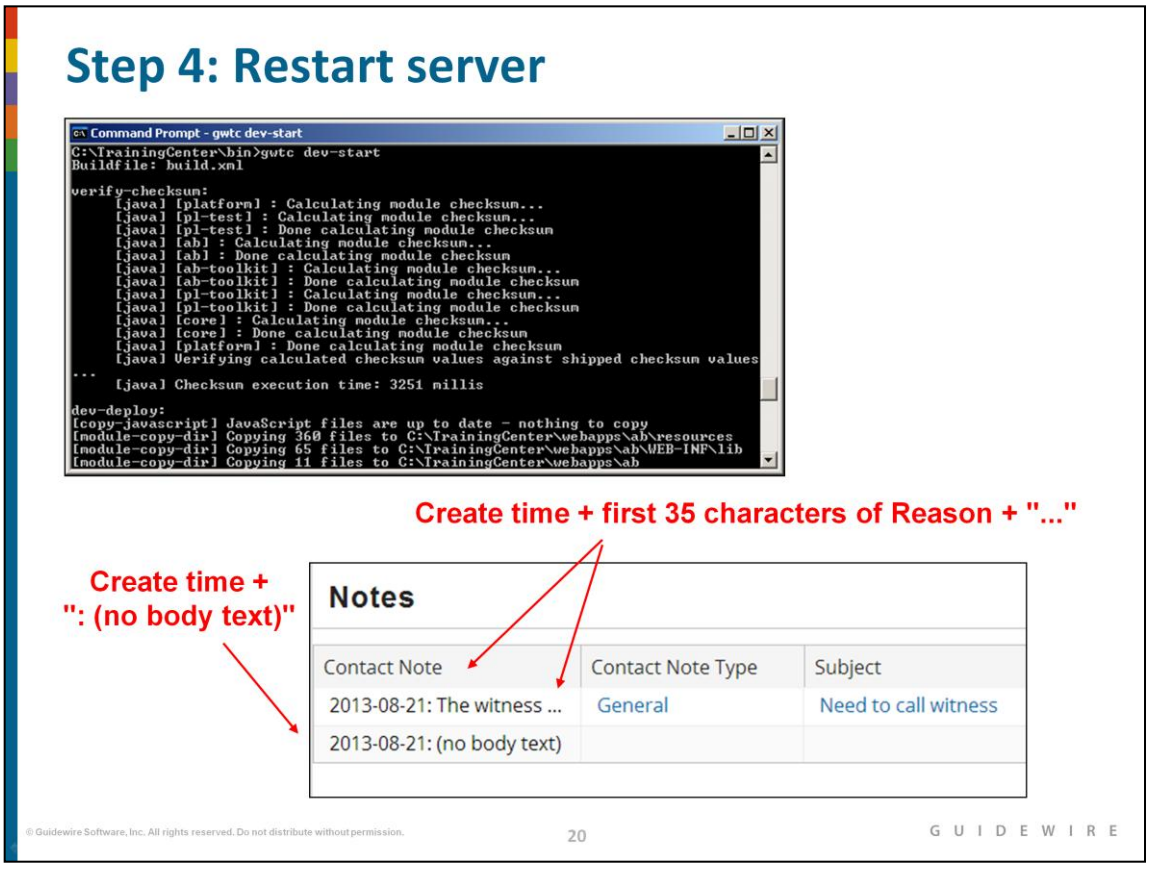
- Code must include return value that returns a string

In the example above, the return value is the CreateTime field plus either:

- A ": " string plus the first 35 characters of the Reason field plus "..." (if the Body is non-null), OR
- A ": (no body text)" string (if the Body is null)



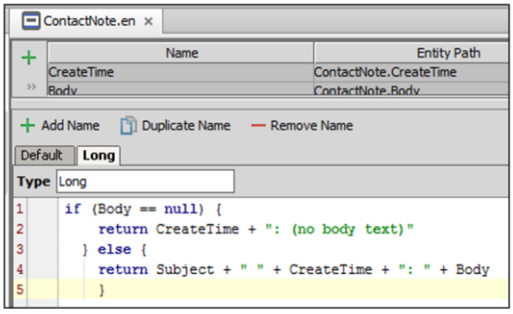
If you have an object named "x", then any UI display reference to "x" is equivalent to "x.DisplayName". Guidewire recommends explicitly adding DisplayName to improve readability of the code.



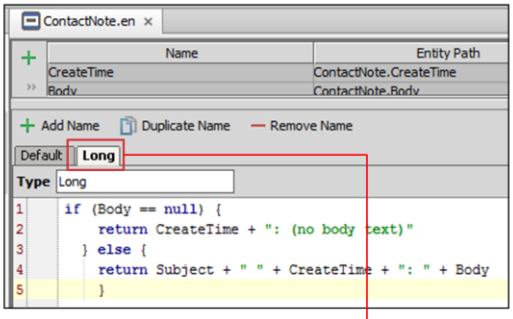
Entity names are inherently tied to the data model. You must therefore restart the server to deploy entity name changes.

Custom display names

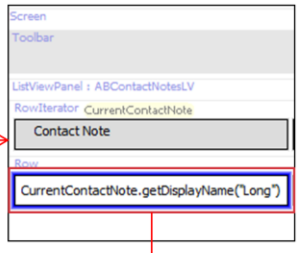
- You can create additional display names
  - Click the "Add Name" button
  - Enter a Type for the additional display name
  - Enter code as usual

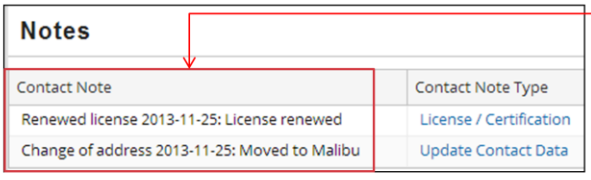


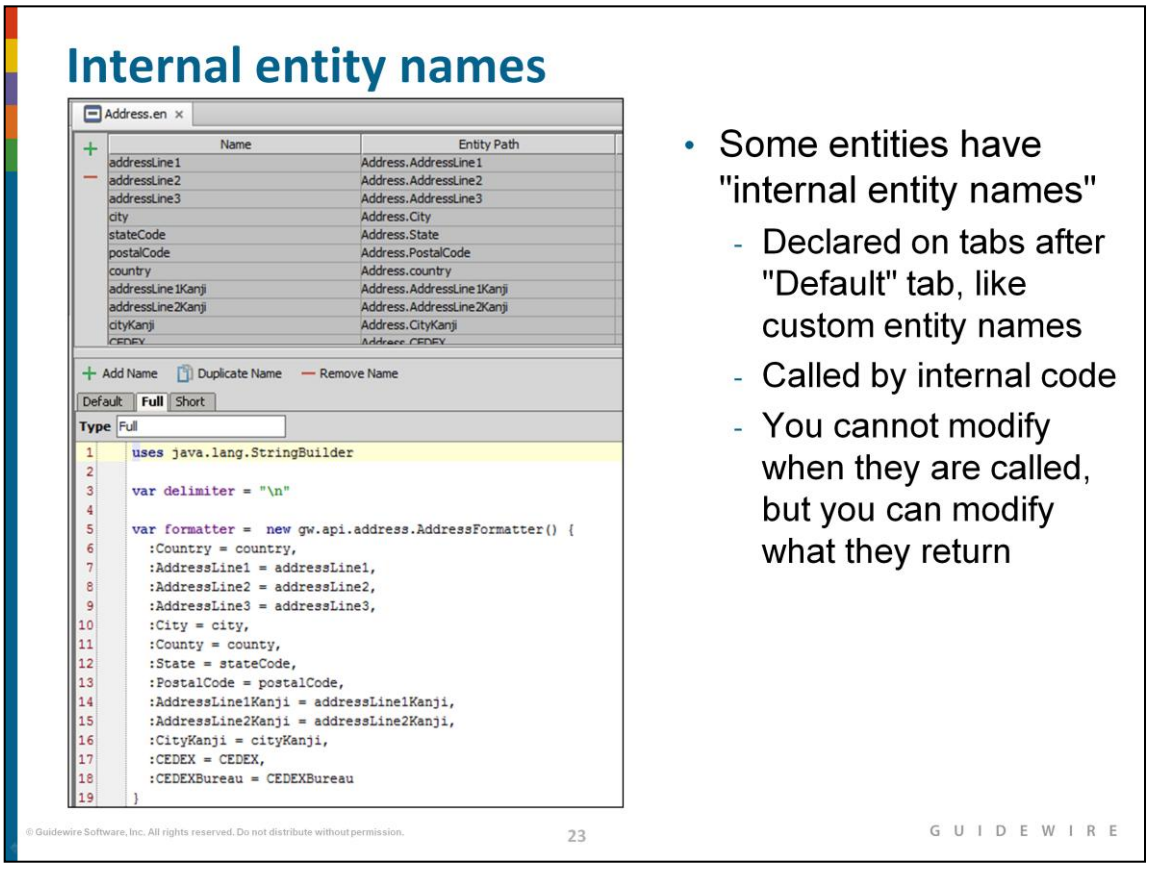
Referencing custom display names



Call the .getDisplayName method using the Type of the custom name as its parameter







For more information about the internal entity names for a given Guidewire application, consult that application's documentation.

This slide shows the AddressFormatter() type, which is a property created for globalization purposes. For information, see the *Globalization Guide* for your application.

Lesson objectives review

You should now be able to:

- Describe the role of entity names
- Create and modify entity names

Review questions

1. For an ABPerson object, name an example of when the object's entity name might appear in the user interface.
2. What field points to an object's entity name?
3. For which entities do you need to create an entity name?
4. What Gosu keyword always appears in the code used to generate an entity name?
5. What must you do to deploy new entity names or changes to existing entity names?

Answers

1. Possible answers: a dropdown that lists the ABPerson in a list of ABPersons; an info bar widget with a value property set to the ABPerson's DisplayName.
2. DisplayName
3. Entities whose display names are referenced in the UI that do not have entity names (typically "secondary" base application entities and custom entities).
4. return
5. You must restart the server.

Notices