

Creating an Entry constructor

By default, when an instance of class is created using an Alf expression of the form `new ClassName()`, each attribute of the new instance has its default value, if this is specified for it, and is null otherwise. However, sometimes it is desirable to perform additional initialization behavior on a newly created instance, beyond just the default initialization. You can do this by defining a *constructor* operation.

A constructor operation is created like any other operation, but it must then have the standard *Create* stereotype applied. A class may have any number of constructors. Conventionally, a constructor often has the same name as the class (which, in Alf, is considered the default), but, in UML, this is not required – constructors can have any name. An Alf instance creation expression identifies a specific constructor to use, with a slightly simplified syntax for the case in which the constructor name is the same as the class name.

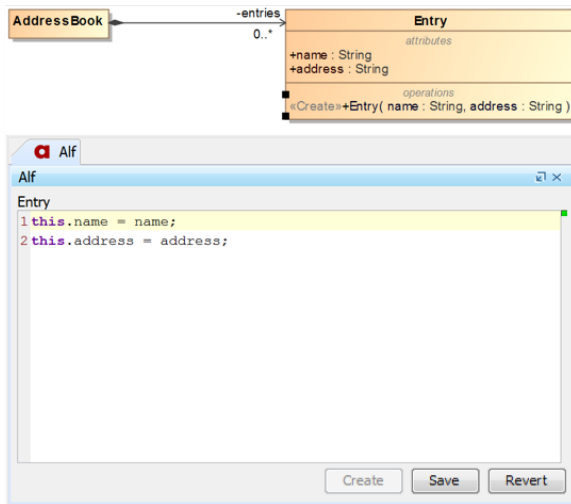
In the Address Book model, it will be useful for the *Entry* class to have a constructor operation, so that values can be provided for the *name* and *address* attributes of a new instance when it is created.

Related pages

- [The Alf editor](#)
- [Using Alf for Operation methods](#)

To create an *Entry* constructor

1. Click on the little **Create Element** button at the top right of the *Entry* symbol on the Class diagram, and select **Operation**.
2. Type *Entry* as the name of the Operation, and, *inside the parentheses*, type the parameter list *name: String, address: String*, then press **Return**.
3. Right click on the new Operation, and select **Stereotype**.
4. Check the **Create** Stereotype, and click on **Apply**.
5. With the *Entry* Operation still selected, open the Alf editor window (select **Windows > Alf**), if it isn't already open, and press **Create**.
6. In the [Alf editor](#), type the statements shown in the following figure.



7. When the text is correct, press **Save**.



If you define an explicit constructor operation for a class, then default construction is no longer allowed. For example, since the *Entry* Class now has an explicit constructor, the expression `new Entry()` is no longer legal. You must use the explicit constructor, with *name* and *address* arguments. However, if desired, you can still define an explicit constructor that looks like a default constructor, having the same name as its class and no arguments. Thus, if you added a constructor *Entry()*, with no Parameters, to the *Entry* Class (overloading the constructor of the same name with Parameters), then it would be legal again to code `new Entry()`, which would call the explicit constructor with no Parameters.

Next: [Creating AddressBook operations](#)