

# PhoneBook.Domain -- learn to declare domain object classes

There is one file for each domain object class:

- `Location.cs`
- `Person.cs`
- `PhoneNumber.cs`

What's more, we have `Country.cs` is with enums for countries used by the `Location` class.

The role of the following files is not explained in this walk-through, this sample code is related to re-store's [Query manager](#), which has become an arcane topic since the introduction of [re-linq](#):

- `queries.xml`
- `QueryManLocation.cs`
- `QueryManPerson.cs`

The role of the `Globalization` folder is explained on page [FIXME](#).

Note that each domain class is derived from `BindableDomainObject`. In the [PhoneBook tutorial](#), the domain classes are derived from `DomainObject` at first, but later this is amended to `BindableDomainObject`. The reason is this. The `DomainObject` class implements all of re-store's piping for persistence, relation automation, etc. `BindableDomainObject`-s can do all that, but also know how to talk to the GUI in a re-motion web application. The difference between `DomainObject` and `BindableDomainObject` is explained here: [bindable domain object](#). For the next few sections you won't need this digression, however. Just remember that `DomainObject` embodies all the nice re-store features we are about to discuss.

See next

- [Properties and property attributes \(Location class\)](#)
- [The DBTable attribute \(Location class\)](#)
- [NewObject and GetObject \(Location class\)](#)
- [Linq and domain objects \(Location class\)](#)