13 Using System.Collections

The .Net framework includes many new data types. The Collections namespce contains types which are useful for creating data types which can dynamically add, remove and manage elements. Creating an array can be done using an ArrayList. Before this will work, you must add:

using System.Collections

To either your project or execute the line in your session.

```
      a = ArrayList()
      a.Add(10)
      a.Add(100)
      a.Count
2
      a.GetType()

System.Collections.ArrayList
```

Another powerful collection is the Hashtable:

```
      a = Hashtable()
      a["test"] = 100 200 300
      a["hello"] = "some text"
      a["test"]
100 200 300
      a["hello"]
some text
```

The newest addition to .Net 2.0, Generics, are also supported. To include generics in your project, place this at the top of your file, or run it in the session:

```
using System.Collections.Generic
```

Generics are collections which can be typed. This provides both an increase in speed and compatibility with other .Net programs. For instance, the Generic Dictionary is similar to the Hashtable shown above, however it is instantiated in a different manner:

```
      a = Dictionary[String, Int32]()

      a
System.Collections.Generic.Dictionary`2[System.String,System.Int32]

      a.Add("more", 10)


      a["more"]

10


      a.Add(10, "more")

bad args for method
```

As you can see, only keys of string type and data of integer type can be added to the collection.