

Fast class factories

Fast class factories

A **class factory** is a special class that constructs other classes on demand, based on external configuration data.

Fast class factories

```
// get database connection  
DbConnection connection =  
    ConnectionFactory.GetConnectionFor("sales data");
```


Fast class factories

```
private object GetClassInstance(string typeName)
{
    return .... ??
}
```


Static solution

```
private object GetClassInstance(string typeName)
{
    switch (typeName)
    {
        case "MyClassType1":
            return new MyClassType1();
        case "MyClassType2":
            return new MyClassType2();
        ....
    }
}
```


Activator solution

```
private object GetClassInstance(string typeName)
{
    return Activator.CreateInstance (Type.GetType(typeName));
}
```


Third solution: delegate

```
private object GetClassInstance_MyClassType1()  
{  
    return new MyClassType1;  
}
```


Third solution: delegate

```
private object GetClassInstance_MyClassType1()  
{  
    // IL instructions  
    newObj MyClassType1  
    ret  
}
```


Third solution: delegate

1. Check dictionary if the delegate has been created already
2. If so → retrieve and call delegate
3. If not →
 4. create dynamic method and write newObj and ret instructions into it.
 5. Wrap method in a delegate and store in dictionary
 6. Call delegate to instantiate class

Fast class factories

A **class factory** is a class that constructs class instances on demand using external configuration information.

- The **Activator** class is **86 times slower** than compiled code.
- **Dynamic method delegates** are **5 times slower** than compiled code and **17 times faster** than the **Activator** class.

Replace the **Activator** class with **dynamic method delegates** in your class factories to speed up your code by **a factor of 17!**