**AutoPoco -** A framework to fluently build test data for Plain Old Clr Object.

**Silverlight version:** <http://autopoco.codeplex.com/SourceControl/network/Forks/davidpadbury/Silverlight>

            // Perform factory set up (ONCE for entire test run)

            IGenerationSessionFactory factory = AutoPocoContainer.Configure(x =>

            {

                x.Conventions(c =>

                {

                    c.UseDefaultConventions();

                });

                x.AddFromAssemblyContainingType<SimpleUser>();

                // Setup DataSource for each SIMPLE property of your type, the COMPLEX type doesn't need to Setup here.

                x.Include<SimpleUser>()

                    .Setup(c => c.EmailAddress).Use<EmailAddressSource>()

                    .Setup(c => c.FirstName).Use<FirstNameSource>()

                    .Setup(c => c.LastName).Use<LastNameSource>()

                    .Setup(c => c.Age).Use<AgeDataSource>(16, 35)

                    .Setup(c => c.SomeString).Use<RandomStringSource>(5, 10);

                // Built-in DataSource under namespace "AutoPoco.DataSources": Use the VS Object Browser to look it up

                // OR http://autopoco.codeplex.com/wikipage?title=DataSourceList&referringTitle=Documentation

                // Setup DataSource for COMPLEX Type property, By Default the factory will AUTOMATICALLY use this

                x.Include<UserRole>()

                    .Setup(c => c.Name).Use<ValueSource<string>>("Guest");

            });

            // Generate one of these per test (factory will be a static variable most likely)

            IGenerationSession session = factory.CreateSession();

            // Prepare Choices of COMPLEX Type Property

            UserRole roleOne = session.Single<UserRole>()

                                    .Impose(x => x.Name, "Guest").Get();

            UserRole roleTwo = session.Single<UserRole>()

                                    .Impose(x => x.Name, "Admin").Get();

            UserRole roleThree = session.Single<UserRole>()

                                    .Impose(x => x.Name, "Standard").Get();

            // Get a single user

            SimpleUser user = session.Single<SimpleUser>().Get();

            // Get a collection of 100 users, Impose(override) SIMPLE & COMPLEX properties

            // 1. Create three roles 2. Create 100 users 3. The first 50 of those users will be called Rob Ashton 4. Next 50 called Luke Smith

            // 5. 25 Random users will have RoleOne 6. A different 25 random users will have RoleTwo

            // 7. And the other 50 users will have RoleThree 8. And set the password on every single user to Password1

            IList<SimpleUser> users = session.List<SimpleUser>(100)

                  .First(50)

                      .Impose(x => x.FirstName, "Rob")

                      .Impose(x => x.LastName, "Ashton")

                  .Next(50)

                      .Impose(x => x.FirstName, "Luke")

                      .Impose(x => x.LastName, "Smith")

                  .All()

                  .Random(25)

                      .Impose(x => x.Role,roleOne)

                  .Next(25)

                      .Impose(x => x.Role,roleTwo)

                  .Next(50)

                      .Impose(x => x.Role, roleThree)

                 .All()

                      .Invoke(x => x.SetPassword("Password1"))

                 .Get();

            foreach (var item in users)  Console.WriteLine(string.Format("{0}:{1}:{2}:{3}:{4}:{5}:{6}", item.FirstName, item.LastName, item.Age,

                    item.EmailAddress, item.SomeString, item.Password, item.Role.Name));

Sample & Supporting Classes

    public class SimpleUser

    {

        public string FirstName { get; set; }

        public string LastName { get; set; }

        public int Age { get; set; }

        public string EmailAddress { get; set; }

        public string SomeString { get; set; }

        public string Password { get; private set; }

        public UserRole Role { get; set; }

        public void SetPassword(string password) { this.Password = password; }

    }

    public class UserRole

    {

        public string Name { get; set; }

    }

    // Custom DataSource: http://autopoco.codeplex.com/wikipage?title=DataSources&referringTitle=Documentation

    public class AgeDataSource : DatasourceBase<int>

    {

        Random random = new Random();

        private int min = 0;

        private int max = 0;

        public AgeDataSource(int min, int max)

        {

            this.min = min;

            this.max = max;

        }

        public override int Next(IGenerationSession session)

        {

            return random.Next(min, max);

        }

    }