

## SERIALISATION D'UN GRAPHE D'OBJET ACTIVE RECORD DANS WCF

### 1. Problématique

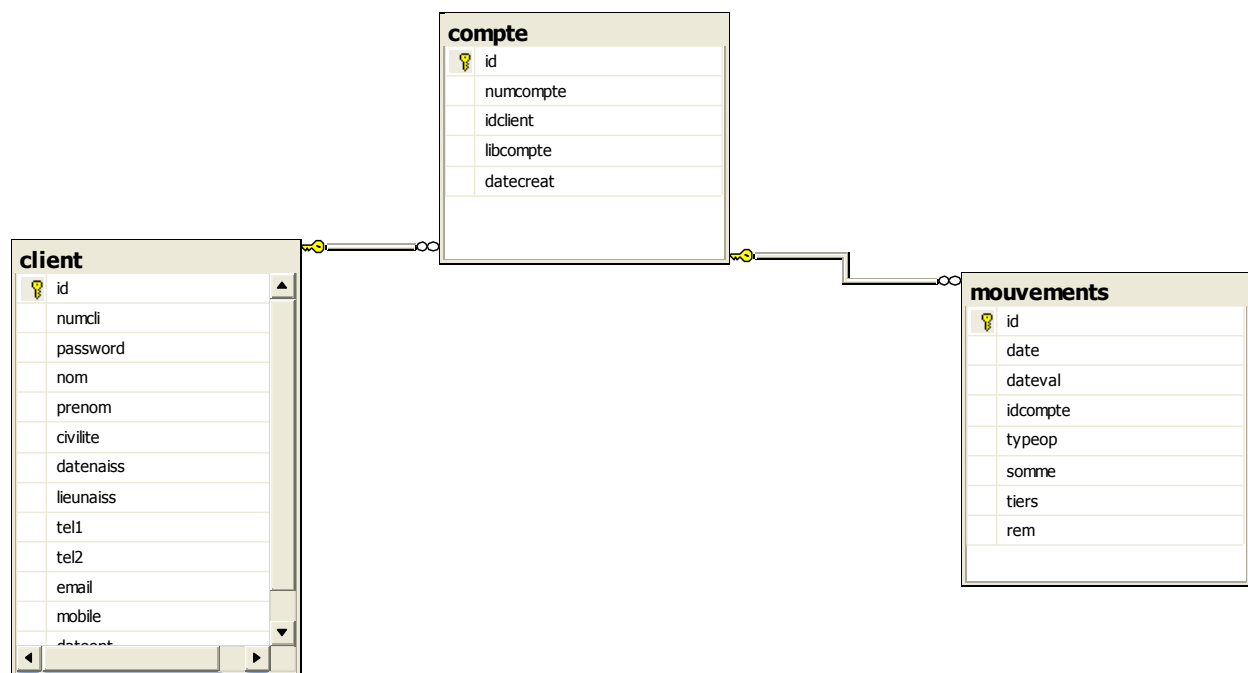
Active record est un pattern implémenté dans les frameworks de type Rails. Certaines produits existent pour .net et Java.

Un Active Record est un POCO (en .NET) dérivant d'une classe de base qui intègre les opérations de base (find, delete, create ...).

Dans la technologie .NET, le framework el plus avancé est Castle (<http://www.castleproject.org>). Ce framework s'appuie sur Nhibernate pour la couche mapping OR.

Par défaut, WCF ne sait pas sérialiser les objets avec des références cycliques ce qui est gênant vis-à-vis du principe de réciprocité des relations du framework Castle (à 1 relation 1-n correspond une relation 1-1).

### 2. Présentation du modèle testé



### 3. Objets du domaine

#### Client.cs

```
using System;
using System.Text;
using System.Collections;
using Castle.ActiveRecord;
using System.Runtime.Serialization;
using System.ServiceModel;

namespace bankdemoDAL
{
    [ActiveRecord("client", Schema="dbo")]
    [DataContract]

```

```

[KnownType(typeof(compte))]
public partial class client : ActiveRecordBase<client> {

private string _nom;
private string _prenom;
private string _civilite;
private System.DateTime _datenaiss;
private string _lieunaiss;
private string _tel1;
private string _tel2;
private string _email;
private string _mobile;
private System.DateTime _dateent;
private long _id;
private string _numcli;
private string _password;
private IList _comptes;

[Property("nom", ColumnType="String", NotNull=true)]
[DataMember]
public string nom {
    get {
        return this._nom;
    }
    set {
        this._nom = value;
    }
}

[Property("prenom", ColumnType="String")]
[DataMember]
public string prenom {
    get {
        return this._prenom;
    }
    set {
        this._prenom = value;
    }
}

[Property("civilite", ColumnType="String")]
[DataMember]
public string civilite {
    get {
        return this._civilite;
    }
    set {
        this._civilite = value;
    }
}

[Property("datenaiss", ColumnType="DateTime", NotNull=true)]
[DataMember]
public System.DateTime datenaiss {
    get {
        return this._datenaiss;
    }
    set {
        this._datenaiss = value;
    }
}

[Property("lieunaiss", ColumnType="String")]
[DataMember]
public string lieunaiss {
    get {
        return this._lieunaiss;
    }
    set {
        this._lieunaiss = value;
    }
}

[Property("tel1", ColumnType="String")]
[DataMember]
public string tel1 {

```

```

    get {
        return this._tel1;
    }
    set {
        this._tel1 = value;
    }
}

[Property("tel2", ColumnType="String")]
[DataMember]
public string tel2 {
    get {
        return this._tel2;
    }
    set {
        this._tel2 = value;
    }
}

[Property("email", ColumnType="String")]
[DataMember]
public string email {
    get {
        return this._email;
    }
    set {
        this._email = value;
    }
}

[Property("mobile", ColumnType="String")]
[DataMember]
public string mobile {
    get {
        return this._mobile;
    }
    set {
        this._mobile = value;
    }
}

[Property("dateent", ColumnType="DateTime")]
[DataMember]
public System.DateTime dateent {
    get {
        return this._dateent;
    }
    set {
        this._dateent = value;
    }
}

[PrimaryKey(PrimaryKeyType.Native, "id", ColumnType="Int64")]
[DataMember]
public long id {
    get {
        return this._id;
    }
    set {
        this._id = value;
    }
}

[Property(ColumnType="String")]
[DataMember]
public string numcli {
    get {
        return this._numcli;
    }
    set {
        this._numcli = value;
    }
}

[Property(ColumnType="String")]
[DataMember]

```

```

public string password {
    get {
        return this._password;
    }
    set {
        this._password = value;
    }
}

[HasMany(typeof(compte), ColumnKey = "idclient", Table = "compte", Inverse=true)]
[DataMember]
public IList comptes {
    get {
        return this._comptes;
    }
    set {
        this._comptes = value;
    }
}
}
}
}

```

## **COMPTE.CS**

```

using System;
using System.Text;
using System.Collections;
using Castle.ActiveRecord;
using System.Runtime.Serialization;
using System.ServiceModel;

```

```

namespace bankdemoDAL
{
    [ActiveRecord("compte", Schema = "dbo")]
    [DataContract]
    [KnownType(typeof(mouvements))]
    [KnownType(typeof(client))]
    public partial class compte : ActiveRecordBase<compte>
    {

        private string _libcompte;

        private System.DateTime _datecreat;

        private long _id;

        private string _numcompte;

        private IList _mouvements;

        private client _client;

        [Property("libcompte", ColumnType = "String")]
        [DataMember]
        public string libcompte
        {
            get
            {
                return this._libcompte;
            }
            set
            {
                this._libcompte = value;
            }
        }

        [Property("datecreat", ColumnType = "DateTime")]
        [DataMember]
        public System.DateTime datecreat
        {
            get
            {

```



```

using System.Runtime.Serialization;
using System.ServiceModel;

namespace bankdemoDAL
{
    [ActiveRecord("mouvements", Schema = "dbo")]
    [DataContract]
    [KnownType(typeof(compte))]
    public partial class mouvements : ActiveRecordBase<mouvements>
    {

        private System.DateTime _date;

        private System.DateTime _dateval;

        private string _typeop;

        private decimal _somme;

        private string _tiers;

        private string _rem;

        private long _id;

        private compte _compte;

        [Property("date", ColumnType = "DateTime")]
        [DataMember]
        public System.DateTime date
        {
            get
            {
                return this._date;
            }
            set
            {
                this._date = value;
            }
        }

        [Property("dateval", ColumnType = "DateTime")]
        [DataMember]
        public System.DateTime dateval
        {
            get
            {
                return this._dateval;
            }
            set
            {
                this._dateval = value;
            }
        }

        [Property("typeop", ColumnType = "String")]
        [DataMember]
        public string typeop
        {
            get
            {
                return this._typeop;
            }
            set
            {
                this._typeop = value;
            }
        }

        [Property("somme", ColumnType = "Decimal")]
        [DataMember]
        public decimal somme
        {
            get
            {

```

```

        return this._somme;
    }
    set
    {
        this._somme = value;
    }
}

[Property("tiers", ColumnType = "String")]
[DataMember]
public string tiers
{
    get
    {
        return this._tiers;
    }
    set
    {
        this._tiers = value;
    }
}

[Property("rem", ColumnType = "String")]
[DataMember]
public string rem
{
    get
    {
        return this._rem;
    }
    set
    {
        this._rem = value;
    }
}

[PrimaryKey(PrimaryKeyType.Native, "id", ColumnType = "Int64")]
[DataMember]
public long id
{
    get
    {
        return this._id;
    }
    set
    {
        this._id = value;
    }
}

[BelongsTo("idcompte", NotNull = false)]
[DataMember]
public compte compte
{
    get
    {
        return this._compte;
    }
    set
    {
        this._compte = value;
    }
}
}
}

```

#### **4. Solution**

Lorsque le DataContractSerializer est créée, il faut positionner le paramètre preserveObjectReferences à true.

Pour cela :

- créer une classe dérivant de `DataContractSerializerOperationBehavior` dans le projet des services WCF
- créer une classe gérant l'attribut qui sera ajouté au contrat (l'interface)

## Contrat du service WCF

```
using System;
using System.ServiceModel;
using System.Runtime.Serialization;
using bankdemoDAL;

namespace bankdemoBL
{
    [ServiceContract()]
    public interface IOperations
    {
        [OperationContract]
        long authentifie(string user, string password);
        [OperationContract]
        ///
        /// Attribut permettant de conserver les références d'objets
        ///
        [PreserveReferences]
        System.Collections.Generic.ICollection<movements> getAllMouvements(long idcompte);
        [OperationContract]
        [PreserveReferences]
        System.Collections.Generic.ICollection<compte> getComptesForClient(long idclient);
        [OperationContract]
        bankdemoDAL.movements getMouvement(long idmouv);
        [OperationContract]
        bankdemoDAL.client getFicheClient(long idclient);
        [OperationContract]
        bankdemoDAL.compte getFicheCompte(long idcompte);
    }
}
```

## Code du service WCF

```
using System;
using System.Collections.Generic;
using System.Collections;
using System.Text;
using bankdemoDAL;
using System.ServiceModel;
using Castle.ActiveRecord.Framework;
using Castle.ActiveRecord.Queries;

[assembly: CLSCompliant(true)]

namespace bankdemoBL
{
    public class Operations : bankdemoBL.IOperations
    {
        public Int64 authentifie(string user, string password)
        {
            IdaoGenerics<client, Int64, compte> dao = new daoGenerics<client, Int64, compte>();
            SimpleQuery query = new SimpleQuery(typeof(client), @"from client c where c.numcli = ? and c.password = ?", user, password);
            client[] rep = (client[])dao.executeQuery(query);
            if (rep.Length == 1)
            {
                if (rep[0].numcli == user && rep[0].password == password)
                    return rep[0].id;
                else
                    return 0;
            }
            else
                return 0;
        }
    }
}
```



```

    }
    public IList<mouvements> getAllMouvements(long idcompte)
    {
        IdaoGenerics<compte, Int64, mouvements> dao = new daoGenerics<compte, Int64, mouvements>();
        compte cmp = dao.getObject(idcompte);
        int i = 0;

        IList<mouvements> lstmouv = new List<mouvements>();
        if (cmp != null)
        {
            for(i=0;i<cmp.mouvements.Count;i++)
            {
                lstmouv.Add((mouvements)cmp.mouvements[i]);
            }
            return lstmouv;
        }
        else
            return null;
    }

    public IList<compte> getComptesForClient(long idclient)
    {
        IdaoGenerics<client, Int64, compte> dao = new daoGenerics<client, Int64, compte>();
        client cl = dao.getObject(idclient);
        int i = 0;
        IList<compte> lstcmpt = new List<compte>();

        if (cl != null)
        {
            try{
                for(i=0;i<cl.comptes.Count;i++)
                {
                    lstcmpt.Add((compte)cl.comptes[i]);
                }

                return lstcmpt;
            }
            catch
            {
                throw (new Exception());
            }
        }
        else
            return null;
    }

    public mouvements getMouvement(long idmouv)
    {
        IdaoGenerics<mouvements, Int64, mouvements> dao = new daoGenerics<mouvements, Int64, mouvements>();
        return dao.getObject(idmouv);
    }

    public compte getFicheCompte(long idcompte)
    {
        IdaoGenerics<compte, Int64, mouvements> dao = new daoGenerics<compte, Int64, mouvements>();
        return dao.getObject(idcompte);
    }

    public client getFicheClient(long idclient)
    {
        IdaoGenerics<client, Int64, compte> dao = new daoGenerics<client, Int64, compte>();
        return dao.getObject(idclient);
    }
}
}
}

```

## **Code de la classe MyDataContractSerializerBehaviour**

```

using System;
using System.Collections.Generic;
using System.Runtime.Serialization;
using System.ServiceModel.Description;
using System.Xml;

```

```

namespace bankdemoBL
{
    public class myDataContractBehaviour : DataContractSerializerOperationBehavior
    {
        public myDataContractBehaviour(OperationDescription operationDescription): base(operationDescription) { }

        public override XmlObjectSerializer CreateSerializer(
            Type type, string name, string ns, IList<Type> knownTypes) {
            return CreateDataContractSerializer(type, name, ns, knownTypes);
        }

        private static XmlObjectSerializer CreateDataContractSerializer(
            Type type, string name, string ns, IList<Type> knownTypes) {
            return CreateDataContractSerializer(type, name, ns, knownTypes);
        }

        public override XmlObjectSerializer CreateSerializer(
            Type type, XmlDictionaryString name, XmlDictionaryString ns,
            IList<Type> knownTypes) {
            return new DataContractSerializer(type, name, ns, knownTypes,
                0x7FFF,
                false,
                true /* preserveObjectReferences */,
                null);
        }
    }
}

```

## **CODE DE LA CLASSE GERANT L'ATTRIBUT**

```

using System;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.ServiceModel.Dispatcher;

namespace bankdemoBL
{
    class PreserveReferencesAttribute : Attribute, IOperationBehavior {
        public void AddBindingParameters(OperationDescription description,
            BindingParameterCollection parameters) { }

        public void ApplyClientBehavior(OperationDescription description,
            ClientOperation proxy) {
            IOperationBehavior innerBehavior =
                new myDataContractBehaviour(description);
            innerBehavior.ApplyClientBehavior(description, proxy);
        }

        public void ApplyDispatchBehavior(OperationDescription description,
            DispatchOperation dispatch) {
            IOperationBehavior innerBehavior =
                new myDataContractBehaviour(description);
            innerBehavior.ApplyDispatchBehavior(description, dispatch);
        }

        public void Validate(OperationDescription description) { }
    }
}

```

## **5. Référence**

1. <http://blogs.msdn.com/sowmy/archive/2006/03/26/561188.aspx>

