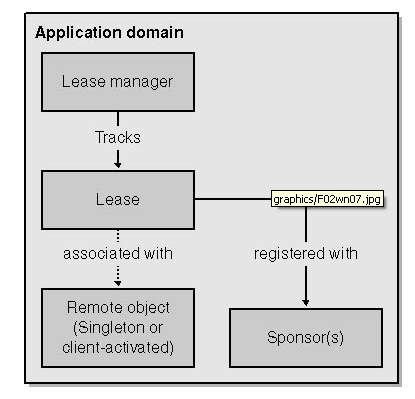
**Notas sobre remoting;**

**Lease time:**

****

**Lease Managers**

Lease managers must periodically examine leases for time expiration. When a lease's time has expired, the lease is informed and it attempts to renew itself by invoking its sponsors.

The lease manager also maintains a list of sponsors from which leases are waiting for replies. If a sponsor does not respond in the interval specified by the **SponsorshipTimeOut** time span, it is removed from the sponsor list.

**Lease**

Es un objeto que encapsula los valores de TimeSpan que la infraestructura de. NET Remoting utiliza para administrar la vida de un objeto remoto.

**Initial lease time o leaseTime**

Especifica el período inicial de tiempo que un objeto permanecerá en la memoria

antes de que el lease mannager de alquiler comience el proceso de eliminar el objeto.

**Sponsors**

Son objetos que pueden renovar contratos de arrendamiento de los objetos remotos.

**ILease properties**

* **CurrentLeaseTime**

Especifica el intervalo de tiempo que queda antes de que el leaseTime expire

Cuando lease expira. Current lease time se setea a al máximo de current lesasetime o RenewOnCallTime

* **RenewOnCallTime (def 2 min)**

Especifica el máximo tiempo al que es seteado CurrentLeaseTime cuando el lease time se renueva.-

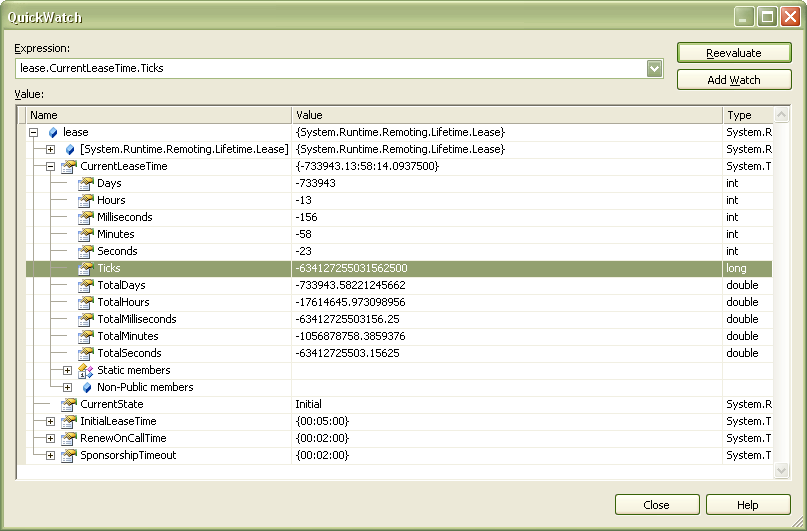
* **SponsorshipTimeout (2 min)**

Tiempo el lease mannager espera para que el sponser responda cuando es notificado de que el lease expiro.-

Si el sponsor no responde en determinado tiempo especificado , el sponsor se elimina y otro sponsor es llamado. Si no hay mas sponsors el lease expira y el garbage collector hace un dispose del objeto remoto.- Si el valor es = 0 el lesase no registra ningún sponsor

* **SponsorshipTimeout**

Indicates how long the lease manager will wait after notifying a sponsor that the lease has expired. We'll look at sponsors shortly.



public override object InitializeLifetimeService()

{

ILease lease = (ILease)base.InitializeLifetimeService();

if (lease.CurrentState == LeaseState.Initial)

{

lease.InitialLeaseTime = TimeSpan.FromMinutes(1);

lease.SponsorshipTimeout = TimeSpan.FromMinutes(2);

lease.RenewOnCallTime = TimeSpan.FromSeconds(2);

}

return lease;

//return base.InitializeLifetimeService();

}

public override System.Runtime.Remoting.ObjRef CreateObjRef(Type requestedType)

{

return base.CreateObjRef(requestedType);

}

**Instancias de proxy:**

S A O Singleton

* Mantiene el estado: Los miembros privados permanecen
* Los valores estáticos
* Muere cada Initial Lease time
* Muere el objeto en el server si otro cliente no lo utiliza

**Orden de ejecución de métodos:**

1. CreateObjRef
2. InitializeLifetimeService

Si pasan InitialLeaseTime minutos.. se muere el objeto remoto y empiesa nuevamente pasos 1 y 2

Mientras se ejecuten metodos o llamadas por parte de los clientes no se mueren los objetos en el server.-