

---

Subject: Re: Syncing or changing BuildingGameObj 'IsDetroyed' state for clients  
Posted by [iRANian](#) on Thu, 01 Aug 2013 14:04:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I tried it, doesn't seem to work. I'm using:

```
void Revive_Building(GameObject *Building)
{
    if (Building == nullptr ) return;
    if ( Is_Building_Dead(Building) == false ) return;

    int Team = ((DamageableGameObj*)Building)->Get_Player_Type();

    Restore_Building(Building);

    Building->Set_Object_Dirty_Bit(NetworkObjectClass::BIT_CREATION, true);
    Update_Network_Object(Building);

    float max = Commands->Get_Max_Health(Building);
    Commands->Set_Health(Building, max);

    // This is needed to update the state of a building from 'dead' to 'alive on the client
    Update_Network_Object(Building);
    Commands->Apply_Damage(Building, 1.0f, "Explosive", 0);
    Update_Network_Object(Building);

    Commands->Apply_Damage(Building, 1.0f, "Explosive", 0);
    Update_Network_Object(Building);

    if (Is_Base_Powered(Team)) { Commands->Set_Building_Power(Building, true); }
    else { Commands->Set_Building_Power(Building, false); }

    GameObject *Ref = Find_Refinery(Team);
    bool RefDead = Commands->Get_Health(Ref) == 0.0f;
    bool HarvDead = Commands->Get_Health(Find_Harvester(Team)) == 0.0f;

    // Console_Output("HarvDead = %d, HarvHealth = %f, RefDead = %d, Ref = %x, Harv = %x",
    HarvDead, Commands->Get_Health(Find_Harvester(Team)),
    // RefDead, Ref, Find_Harvester(Team)); // DEBUG CRAP

    // Build a new Harvester if needed
    if ( HarvDead && !RefDead )
    {
        Request_New_Harvester(Team);
    }

    // Needed for re-initialisation and also updating the state of the buidling
    ((BuildingGameObj*)Building)->Collect_Building_Components();
```

```
Update_Network_Object(Building);
```

```
Update_Building_State(Building, false);  
Update_Network_Object(Building);
```

```
Initialize_Building(Building);  
Update_Network_Object(Building);
```

```
auto c = BaseControllerClass::Find_Base(Team);  
c->Set_Object_Dirty_Bit(NetworkObjectClass::BIT_RARE, true);  
Update_Network_Object(c);  
}
```

I call Restore\_Building() which is a scripts 4.0 API function which does the following:

```
void SCRIPTS_API Restore_Building(GameObject* obj)  
{  
    if (!obj) return;
```

```
    BuildingGameObj* building = obj->As_BuildingGameObj();  
    if (!building || !building->Is_Destroyed())  
        return;
```

```
    building->Set_Is_Destroyed(false);
```

```
    BaseControllerClass* base = BaseControllerClass::Find_Base(Get_Object_Type(building));  
    if (base)
```

```
    {  
        if (building->As_SoldierFactoryGameObj())  
            base->Set_Can_Generate_Soldiers(true);
```

```
        if (building->As_VehicleFactoryGameObj())  
            base->Set_Can_Generate_Vehicles(true);
```

```
        base->Set_Object_Dirty_Bit(NetworkObjectClass::BIT_RARE, true);  
    }  
}
```