WeaponBagClass functions:

```
 WeaponClass *  Add_Weapon( const WeaponDefinitionClass * def, int rounds = 0, bool
give_weapon = true );
 WeaponClass *  Add_Weapon( int id, int rounds = 0, bool give_weapon = true );
 WeaponClass *  Add_Weapon( const char *weapon_name, int rounds = 0, bool give_weapon =
true );
```

You can get a WeaponBagClass pointer for an GameObject like:

```
SCRIPTS_API const wchar_t *Get_Current_Wide_Translated_Weapon(GameObject *obj)
{
 if (!obj)
 {
  return 0;
 }
 PhysicalGameObj *o = obj->As_PhysicalGameObj();
 if (!o)
 {
  return 0;
 }
 ArmedGameObj *o2 = o->As_ArmedGameObj();
 if (!o2)
 {
  return 0;
 }
 WeaponBagClass *w = o2->Get_Weapon_Bag();
```

SoldierGameObj::Give_All_Weapons(void) loops through weapon definitions and grants everyone of them to a soldier withWeaponBagClass::Add_Weapon(char const*,int,bool). It loops through all definitions with DefinitionMgrClass::Get_First(ulong,DefinitionMgrClass::ID_TYPE) and call DefinitionMgrClass::Get_Next(DefinitionClass *,ulong,DefinitionMgrClass::ID_TYPE). It sets the amount of ammo for the granted weapon to -1 and passes 'true' to the 'give_weapon' bool arg. Those two DefinitionMgrClass functions return a DefinitionClass object (NOT pointer), and DefinitionClass::Get_Name() is probably called to get the weapon name to pass to Add_Weapon().