
Subject: scripts.dll 1.8 progress report

Posted by [jonwil](#) on Tue, 20 Jul 2004 10:09:31 GMT

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

So far, I have implemented the following pieces of code (none of which are in any way based on anything SK has written)

- 1.InitEngine (this sets up the different addresses for stuff for game.exe, server.dat, RH7 FDS and RH8 FDS depending on what is needed)
 - 2.code to detect the difference between game.exe and server.dat (it works great too)
 - 3.As_PhysicalGameObj (will return the passed in object if it is a PhysicalGameObj, otherwise 0)
 - 4.As_DamageableGameObj (will return the passed in object if it is a DamageableGameObj, otherwise 0)
 - 5.As_SoldierGameObj (will return the passed in object if it is a SoldierGameObj, otherwise 0)
 - 6.Get_Model (returns the name of the model file used for the object if it is a valid PhysicalGameObj otherwise the string "DUMMY" is returned, this is the opposite of the Set_Model script command)
 - 7.Get_Object_Type Returns the type of an object, works for any DamageableGameObj
 - 8.Get_Definition (returns the definition class for any GameObject)
 - 9.Get_Phys_Definition (returns the physics definition class for any PhysicalGameObj)
 - 10.Get_Class_ID (returns the class ID for any DefinitionClass)
 - 11.Get_Definition_Name (returns the Definition Name for any DefinitionClass, this matches up with the preset name for at least any DefinitionClass from a GameObject)
 - 12.Get_Definition_ID (like Get_Definition_Name but gets the definition ID, this will match the preset ID for any DefinitionClass from a GameObject)
 - 13.Soldier_Re_Init (this one calls SoldierGameObj::Re_Init to load in a new SoldierGameObjDef)
 - 14.Find_Named_Definition (this one finds a valid definition such as a vehicle, soldier or whatever)
 - 15.Post_Re_Init (this one does stuff that needs to be done after calling Soldier_Re_Init)
 - 16.Change_Character (this one takes a GameObject and a preset name. If the preset doesn't exist or is not a SoldierGameObjDef, it returns false. It will also return false if the GameObject is not a SoldierGameObj. Otherwise, it will find the definition with Find_Named_Definition, call Soldier_Re_Init and Post_Re_Init then return true)
 - 17.WideCharToChar (this one is a utility function that converts a wchar_t *to a char *)
 - 18.Get_Player_Name (this one converts a GameObject into a player name)
 - 19.Get_Player_ID (this one converts a GameObject into a player ID)
 - and 20.Get_GameObj (this one converts a player ID into a GameObject)
-