

# Adding Working Scripts to Items

## Contents

### What it does:

This tutorial will go step-by-step to adding a simple script to an item-- gloves that grant you the Blizzard spell while wearing them.

### Create the item

First, you want to create the gloves. Let's say our gloves are Elven Gauntlets of Thalmor origin. The fields I edited are highlighted in red.

### Writing the script

Now that the easy part is done, we will move on to the difficult part. From Gameplay at the top select [Papyrus Script Manager](#), right click on a script and select New. You will be greeted with a dialogue box. (You could also add the script from the item window by pressing "Add" on the script section and choosing [New Script], that way it will default to the right type)

Because it's an item, it must extend [ObjectReference](#). Because [OnEquipped](#) is an event that will be called natively for ObjectReferences, we will be able to use it in this script.

Now we can simply write out our script and save it.

This is the ScriptName declaration, and its documentation comment:

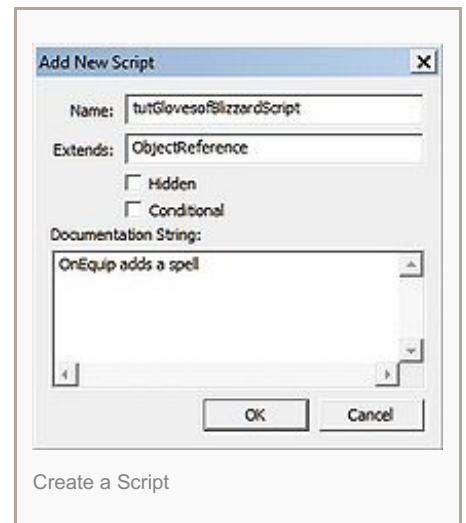
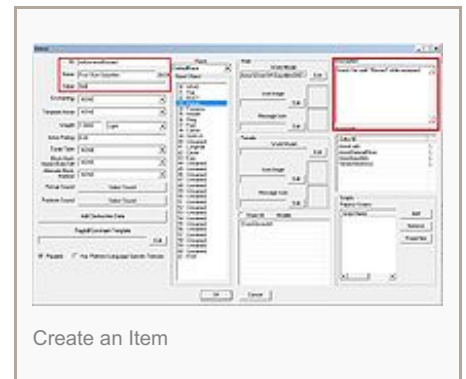
```
ScriptName tutGlovesofBlizzardScript extends
ObjectReference
{OnEquip adds a spell}
```

In order to use things like [AddSpell](#) and [AddItem](#), you have to make a [property](#) for them in your script. If you don't do this, you will get an "undefined variable" error.

```
Spell Property Blizzard auto
```

This next snippet our OnEquipped event. It is important, if this event is being used, that the script extends [ObjectReference](#), or another script that extends [ObjectReference](#) (like [Actor](#)), or else the the event will never fire.

```
Event OnEquipped(Actor akActor)
    If (akActor == Game.GetPlayer())
        Game.GetPlayer().AddSpell(Blizzard)
```



```
EndIf
EndEvent
```

This is simply another event, [OnUnequipped](#) using the same property to remove the spell when the player unequips our scripted gloves. We only want them to have the spell if they wear it.

```
Event OnUnequipped(Actor akActor)
    If (akActor == Game.GetPlayer())
        Game.GetPlayer().RemoveSpell(Blizzard)
    EndIf
EndEvent
```

Now that our script is finished, we can save it. If you've done everything correctly, yours should compile as flawlessly as this example.

## Attaching the Script

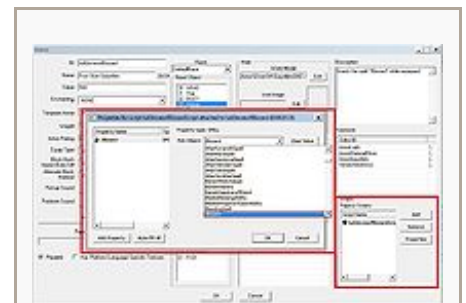
Now, we go back to our gloves in the Creation Kit, and add the script, edit the property and select the spell that will stand in for the Blizzard property.

## Testing In Game

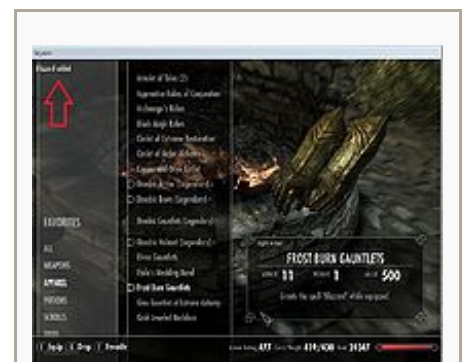
Now that our script has been set up in the Creation Kit, we can test it. I added the gloves in to the game by allowing the players to combine normal elven gauntlets with Frost Salts at the Forge. You can use any method you want to put them in the game, including the console (use the [Help](#) command to find the right formID, and [player.AddItem](#) to add them to the player).

Everything looks good on the gauntlets so far, and when we equip them our script runs successfully.

This concludes the tutorial, and I hope it helped people avoid a lot of frustration.



Attach the Script



View Ingame