

World Builder Basic Scripting Tutorial

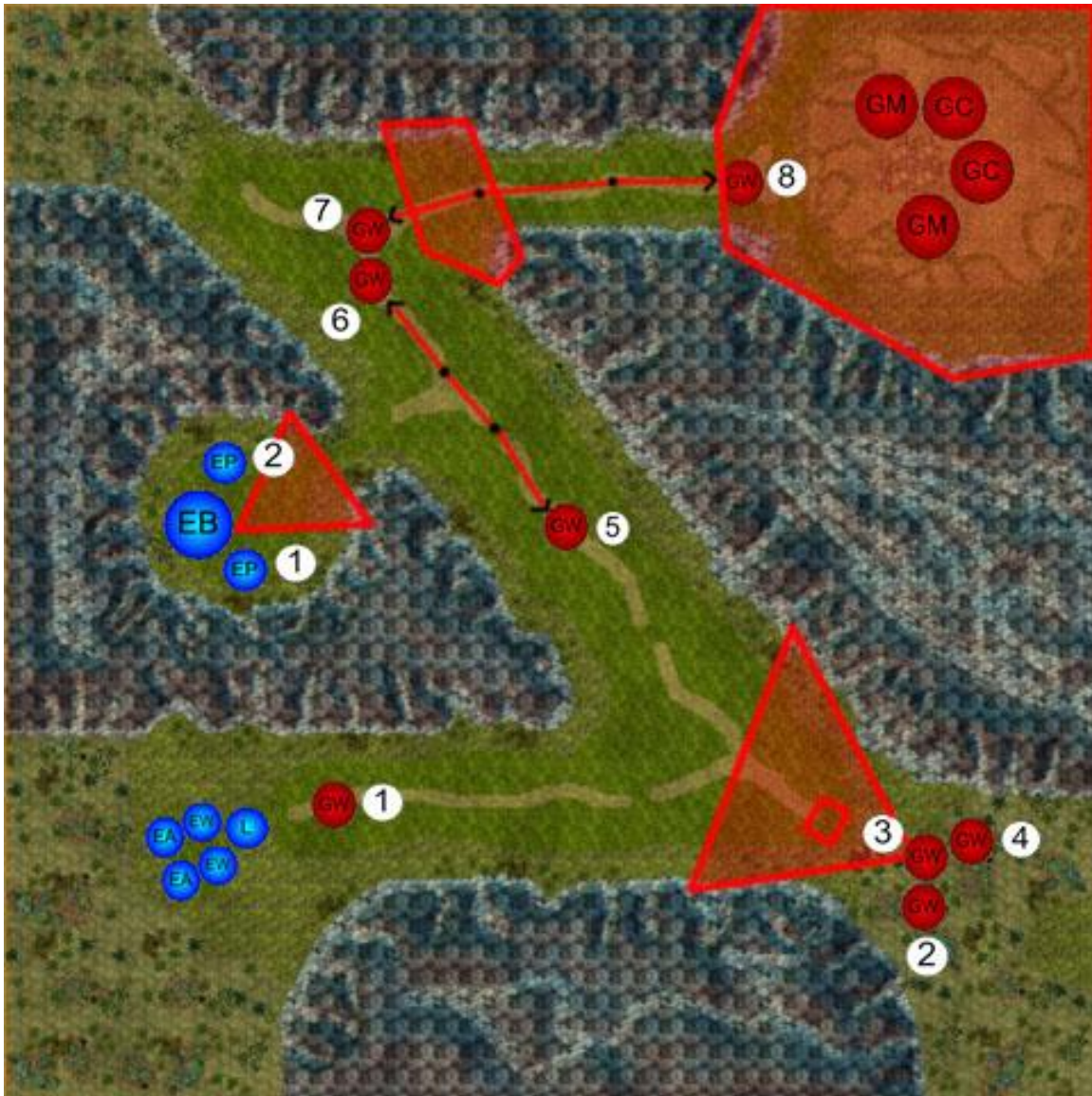


Table of Contents

<i>Table of Contents</i>	2
<i>Introduction</i>	3
<i>Map Layout</i>	4
<i>Team Building</i>	6
<i>Map.str & Map.ini</i>	12
<i>Scripting</i>	14
Intro Cine	22
Objectives	24
Mission Win & Loss	25
PlyrElves Scripts	25
PlyrCivilian Scripts	26
Goblin Base	26
Goblin Patrols	28
PlyrWild Scripts	30

Introduction

Welcome to the Basic Scripting Tutorial for World Builder. While this tutorial will serve to familiarize you with the scripting tool used in World Builder, it won't teach what every script action and condition is. This tutorial will cover the most commonly used scripts that the designers use for The Battle for Middle-earth 2. This tutorial also will assume that you've read up on how to create a map; adjusting the height of the terrain, texturing, placing units, etc. This tutorial will also list every single script needed to recreate this level. Below is a picture of the map that's going to be referenced for this tutorial.



Map Layout

So first lay out the map above in World Builder. Here's a quick list of the units on the map:

- (L) – Legolas
- (EW) – Elven Warrior Horde
- (EA) – Elven Archer Horde
- (EP) – Elven Pike Horde
- (EB) – Elven Barracks
- (GW) – Goblin Warrior Horde
- (GM) – Goblin Mine Shaft
- (GC) – Goblin Cave

In this mission the Player starts with two Elven Warrior Hordes, two Elven Archer Hordes, and Legolas. The player is instructed to clear out all the Goblins in the region. Before you start scripting however, we have to make sure to get all the units named and on their appropriate teams. Also, we'll have to setup a "map.ini" file and a "map.str" file. First, let's go over all the named units and teams you need to have on the map. Referencing the map above, the names and teams they are on are as follows:


- | | |
|----------------------------------|---------------------------------------|
| - (GW)1 - "Goblin Retreater" | Team - PlyrCivilian/teamPlyrCivilian |
| - (GW)2 - No Name | Team - PlyrWild/Ambush Team |
| - (GW)3 - No Name | Team - PlyrWild/Ambush Team |
| - (GW)4 - No Name | Team - PlyrWild/Ambush Team |
| - (GW)5 - No Name | Team - PlyrWild/Goblin Patrol Team 01 |
| - (GW)6 - No Name | Team - PlyrWild/Goblin Patrol Team 02 |
| - (GW)7 - No Name | Team - PlyrWild/Goblin Patrol Team 03 |
| - (GW)8 - No Name | Team - PlyrWild/Goblin Patrol Team 04 |
| - (GM)'s - No Name | Team - PlyrWild/teamPlyrWild |
| - (GC)'s - No Name | Team - PlyrWild/teamPlyrWild |
| - (EP)1 - "Elf Reinforcement 01" | Team - PlyrCivilian/teamPlyrCivilian |
| - (EP)2 - "Elf Reinforcement 02" | Team - PlyrCivilian/teamPlyrCivilian |
| - (EB) - "Elf Barracks" | Team - PlyrCivilian/teamPlyrCivilian |

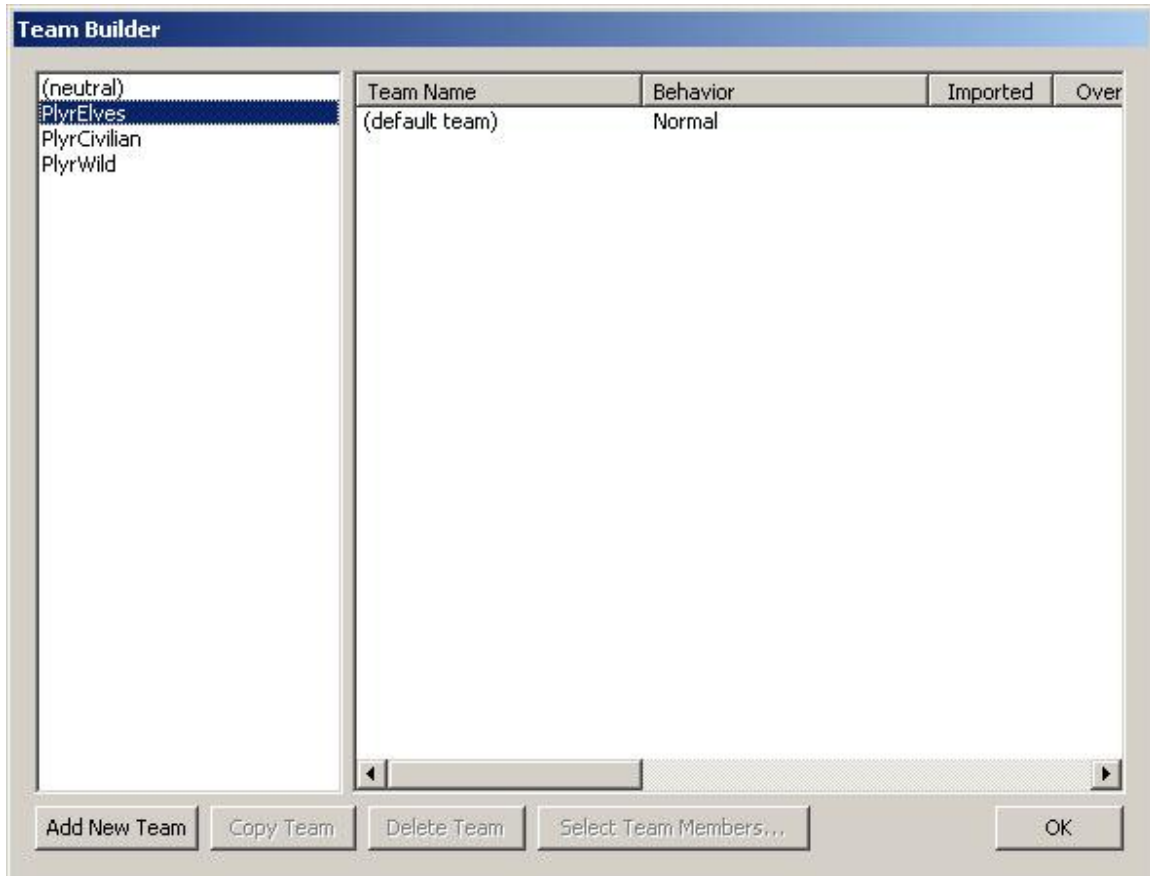
The next thing you'll want to do is lay down any waypoints, area triggers, and waypoint paths you'll need. Here's a quick list of the names, and locations for all these parameters:

- **Waypoint:**
 - o **Location:** Right behind Legolas
 - o **Name:** InitialCameraPosition
- **Waypoint:**
 - o **Location:** Underneath (GW)1
 - o **Name:** WP - Look At Goblin Retreater
- **Waypoint:**
 - o **Location:** Center of small square to left of (GW)3
 - o **Name:** WP - Goblin Retreater Run Here

- **Area Trigger:**
 - o **Location:** Right corner of the screen; Large, triangular
 - o **Name:** AT - Ambush Team Attack
- **Area Trigger:**
 - o **Location:** Inside large triangular Area Trigger on bottom right; Small, square
 - o **Name:** AT - Transfer Goblin Retreater to Wild Player
- **Area Trigger:**
 - o **Location:** Next to (EP)1, (EP)2, and (EB); Rectangular
 - o **Name:** AT - Transfer Reinforcement Elves
- **Area Trigger:**
 - o **Location:** Next to (GW)6 and (GW)7; Make sure it covers the chokepoint
 - o **Name:** AT - Build Goblin Defense and Attack Teams
- **Area Trigger:**
 - o **Location:** Surrounding Goblin base in top right: Make sure it covers base and then some.
 - o **Name:** AT - Goblin Base
- **Waypoint Path:**
 - o **Location:** Going from (GW)5 to (GW)6; One way
 - o **Name:** WP Path - Patrol Path 01
- **Waypoint Path:**
 - o **Location:** Going from (GW)6 to (GW)5; One way
 - o **Name:** WP Path - Patrol Path 02
- **Waypoint Path:**
 - o **Location:** Going from (GW)7 to (GW)8; One way
 - o **Name:** WP Path - Patrol Path 03
- **Waypoint Path:**
 - o **Location:** Going from (GW)8 to (GW)7; One way
 - o **Name:** WP Path - Patrol Path 04

Team Building

And the final step you need is to setup all the teams you'll need for this map. To open up the teams window, click on this button  near the top of your World builder window and it will bring up this screen:

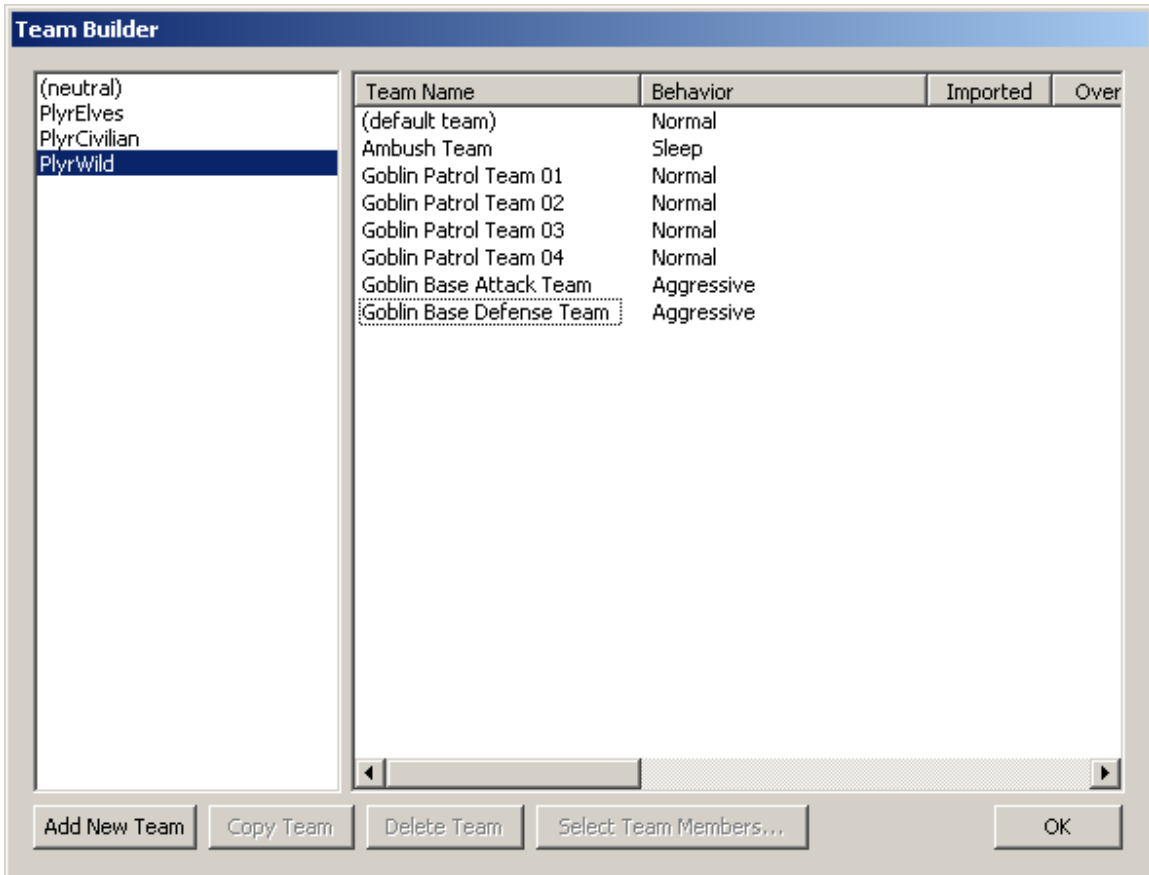


At this point, you ought to have all your Players setup, shown in the left most window of the **Team Builder**. But let's go over what this window is and how it works as well as all the buttons it has and what they do:

- **Left Window** - This window contains any Players you have in your level. Clicking on one of these players will show you all the teams they contain. Each player (except **(neutral)**) will have a "(default team)". This is the team that any new units you put down will join.
- **Right Window** - This window shows all the teams currently created for the Player you have selected. It also contains other tabs showing information related to those teams.
- **Add New Team** - This will bring up the "Edit Team Template" window.
- **Copy Team** - This will create a copy of any team you currently have selected.
- **Delete Team** - This will delete any team you currently have selected. The "(default team)" can never be deleted.

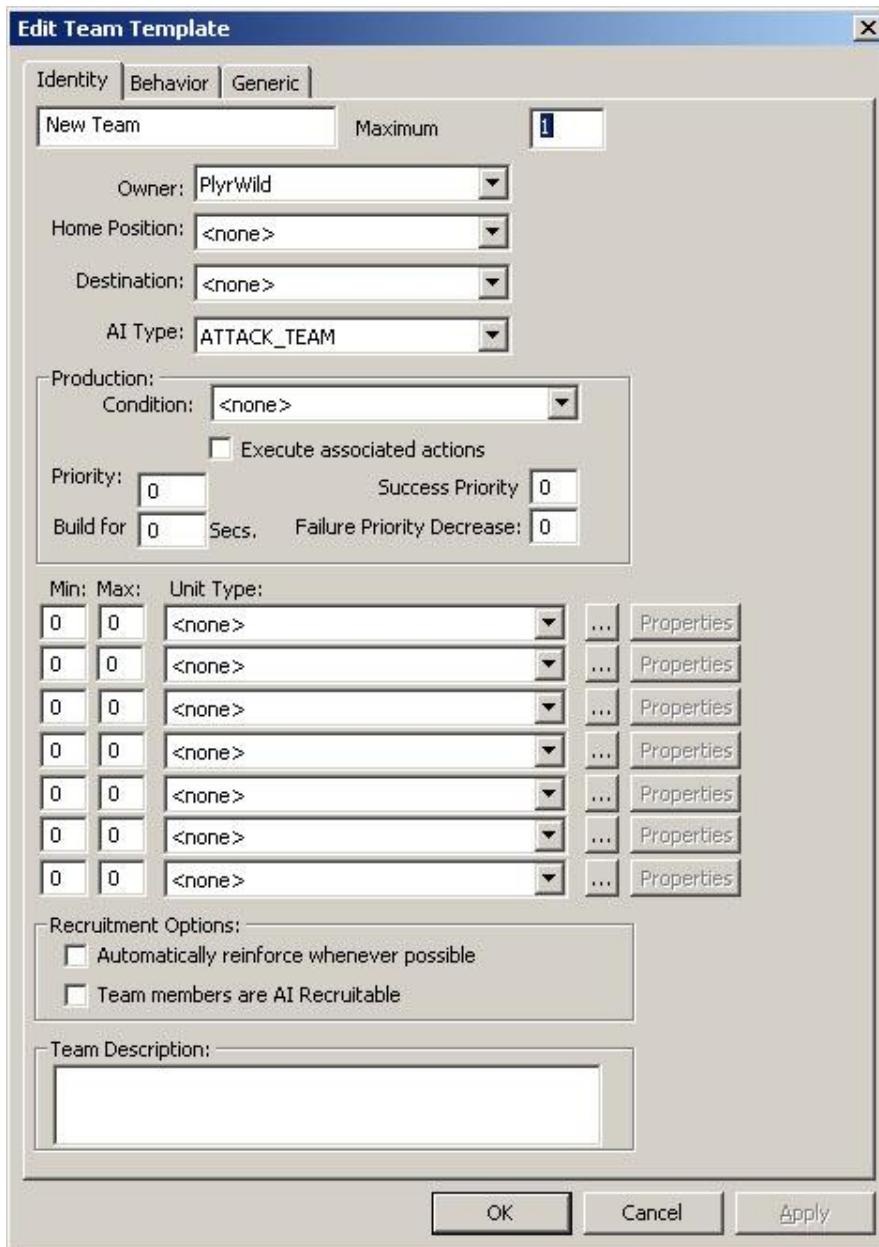
- **Select Team Members...** - This will select any pre-placed units on the map that you have put onto the team you currently have selected.
- **OK** - This will save all the changes you've made and close the **Team Builder** window.

For this mission, the only teams that ever needed to be setup were teams for PlyrWild. Select your PlyrWild and create the teams shown below:



You'll notice that the team's "Behavior" field differs from one team to the next. Also, there are scripts associated with some of the teams as well as unit makeup. So let's go over how to create a team.

The picture below shows us what our brand new team template looks like:

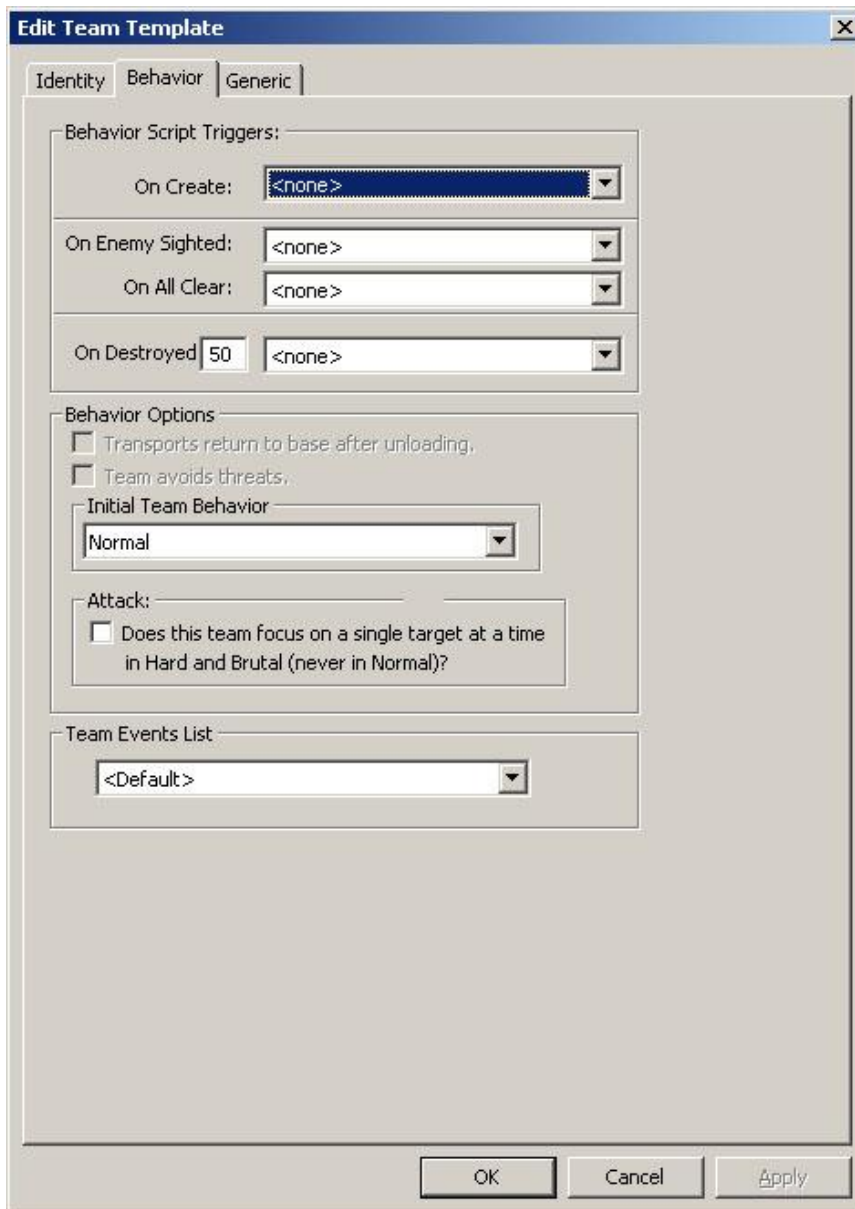


As you can see, there are a ton of fields on the first tab of this window, so let's go over what they all do:

- **New Team** - This little window is actually where you name your team. The default name will always be "New Team" followed by a number in case "New Team" already exists under this player.
- **Maximum** - This is the maximum number of instances of this team the AI will try to have active at all times. For example, if this number is set to 3, then the AI will build 3 instances of this team. As soon as one instance of this team dies, the AI will try to build another instance to make sure it always has 3 active instances.
- **Owner:** - This shows you the current Player that this team belongs to. It's also how you transfer a team from one Player's list to another. Simply open the drop down list and pick which player you wish to have this team belong to.

- **Home Position:** - From this dropdown, you can choose a waypoint which will act as a rally point for this team. Basically, it's where the team will go when it's finished building. Once there, it will execute any scripts attached to it.
- **Destination:** - This field no longer functions. This was carried over from previous versions of World Builder and is no longer used.
- **AI Type:** - This field is no longer useful as our AI Skirmish teams are now handled through code and are no longer scripted. This was useful for the BFME1 Skirmish teams as they were all scripted.
- **Production:**
 - **Condition:** - This dropdown will allow you to choose from a list of scripts that have been marked with the "Subroutine" checkbox. What this field does, is it allows you to create a script with just a condition in it. Then this team will check that script condition and once that condition becomes true, then the AI will start to build this team. This basically allows you to attach a "build me" condition to a team.
 - **Execute associated actions** - When this is checked, any script actions within the attached script from the **Condition** field will fire off as well.
 - **Priority:** - This prioritizes teams within a Player. The higher this number is, the more important this team is, ie: $100 > 1$. This is also how teams recruit from one another. If a team is told to build and has a higher priority than another team that's already been built, then the team that's currently building will try to recruit from the lower priority team.
 - **Build for:** - This number designates how long the AI will try to build this team for before it gives up trying to build it. We usually set this number to 300 seconds (5 minutes). If you leave this number as 0, then your team will probably never build.
 - **Success Priority:** - This was used by our BFME1 AI and is no longer used.
 - **Failure Priority Decrease:** - This was used by our BFME1 AI and is no longer used.
- **Min: Max:** - These fields represent the minimum number and maximum number of units to build of their associated "**Unit Type**". However, the code for these numbers seem to ignore the **Min** field and will always try to build the **Max** field, so when we setup our teams, we always keep these values the same.
- **Unit Type:** - This is a dropdown list of all the units in the game. This is where you compose the makeup of what units your team will build.
- **[...]** - This will bring up a new window to pick your units from. It will fill in the **Unit Type** field for you. It's basically an alternate way to select a unit.
- **Properties** - This will allow you to setup upgrades and experience levels for the units you select in the fields to the right of it.
- **Recruitment Options:**
 - **Automatically reinforce whenever possible** - If this box is checked, this team will always try and recruit members to its team when trying to build.
 - **Team members are AI Recrutable** - If this box is checked then units in this team can be recruited to other, higher priority teams.
- **Team Description** - Here you can write a brief description of what this team does, or how it acts, or any other things you want to remember about this team. This also helps when others are looking at your mission.

The next tab over is the **Behavior** tab. This is where the Team Template starts to interact with scripts more. Let's go over what all the buttons do on this screen:



- **Behavior Script Triggers:**
 - **On Create:** - From this drop down you can choose any script that has been marked with the **Subroutine** checkbox. Then when the code considers this team to be created, it will fire the script in this field. For example, if I wanted this team to go guard an area as soon as it's created, I would create a script telling this team to guard that area and then assign it to this field.
 - **On Enemy Sighted:** - As soon as this team sights an enemy, it will execute the script entered into this field.

- **On All Clear:** - After this team has sighted an enemy and then the enemy goes away, it will execute the script entered into this field. This parameter is sketchy, so it may not work all the time.
- **On Destroyed []** - The number associated with this field is the percentage of health in the team. So if it's 50, then at 50% team health, the team will execute the script entered into this field.
- **Behavior Options**
 - **[] Transports return to base after unloading** - This is left over from Generals and no longer functions.
 - **[] Team avoids threats** - This is left over from Generals and no longer functions.
 - **Initial Team Behavior:**
 - **Peaceful** - The team may be attacked but will not attack back. It will still accept script commands.
 - **Sleep** - This team is basically inactive. It will not accept any script orders, save for a change in **Behavior**, nor will it react to being attacked. We use this to optimize pre-placed teams on the map so they don't eat up framerate.
 - **Passive** - This will reduce the team's sight range so much so that it will only attack enemies that are right on top of it.
 - **Normal** - Default team behavior.
 - **Alert** - Team's sight range is increased by 10% and the team, if following a path, will leave its path to engage any enemies that it detects.
 - **Aggressive** - Team's sight range is increased by 50% and the team, if following a path, will leave its path to engage any enemies that it detects.
 - **Attack:**
 - **[] Does this team focus...** - This is used in case you want a hordes members, or all of the team to focus on a single target, 1 unit not 1 horde, at a time. Essentially this is useful for having the team focus all of its firepower on to one target at a time.
- **Team Event List** - This is leftover from the BFME1 AI and is no longer used.

The final tab on the **Edit Team Template** window is the **Generic** tab. In this window you may put generic scripts into each field. As soon as you populate the first field with a script, a new field will appear under it and so on for every extra script you add. What defines a generic script is that it can be used on multiple teams, units or players. Basically it's a script that calls no specific name, just a generic name such as "<This Team>" or "<All Players>". For example, you could have a script that says, "IF <This Team> enters area trigger "AT - Guard Here", THEN Add 1 to counter "Counter - Guard Teams"". In the example its checking for "<This Team>" which means that it considers "<This Team>" to be whatever team this script is firing from.

Map.str & Map.ini

Now that you have all the map work done, you'll want to create a "map.ini" file and a "map.str" file within the same folder as your .map file for you level. Normally the information that you store in the "map.str" file would go into a much larger file called "lotr.str". If you can unpack and edit that file, then that would be a better place to put this information. You can open the .ini file or the .str file with notepad, they're basically just text files.

The "map.str" contains all the text that you'll want to display in your level. It also contains the text for any objectives you want to display on the objectives screen. Here's what the "map.str" file I created contains:

```
SCRIPT:WBTutorialIntroText_01
```

```
"We've had reports of Goblins in this area. \n We must find them and flush them out."  
END
```

```
SCRIPT:WBTutorialIntroText_02
```

```
"There's some of the scum right there!"  
END
```

```
SCRIPT:WBTutorialIntroText_03
```

```
"They're getting away, after them!"  
END
```

```
SCRIPT:WBTutorialObjectiveText_01
```

```
"Chase after the Goblins."  
END
```

```
SCRIPT:WBTutorialObjectiveText_02
```

```
"Kill the ambush."  
END
```

```
SCRIPT:WBTutorialObjectiveText_03
```

```
"Find the Goblin base and destroy it."  
END
```

```
SCRIPT:WBTutorialBonusObjectiveText_01
```

```
"Find the reinforcements."  
END
```

```
SCRIPT:WBTutorialHintText_01
```

```
"An ambush! Elves, attack!"  
END
```

```
SCRIPT:WBTutorialHintText_02
```

```
"There's more of them than I thought. They must \n have a base around here somewhere."
```

END

```
SCRIPT:WBTutorialHintText_03  
"A patrol, we must be getting close."  
END
```

```
SCRIPT:WBTutorialBonusObjectiveComplete_01  
"You have rallied the Elven reinforcements."  
END
```

The first line in a script to display text is the name of the script, “SCRIPT:WBTutorialIntroText_01”, this is what you’ll see when searching for the “Localized String” variable in World Builder. The next line is what you want to display. And always finish each script with an “End”, otherwise the game will yell at you and may crash out. Here are some more rules for how these entries are setup:

- Always preface the name with “SCRIPT:” then finish with the level it belongs to and what it acts as; Hint, Objective, Intro, etc.
- For any text that you’re going to display in Letterbox mode, put a “\n”. This is how we denote a carriage return in the sentence.

The easiest thing to do is just copy what’s above and paste it into a “map.str” file of your own. Also, in order for World Builder to recognize these string variables, you must have it closed before you save this file. Then when you open it back up, it will read this file and add these variables to its lists.

The next file to make is the “map.ini” file. This file contains your objectives and it also allows you to edit any parameter for any unit in the game. Here’s how the “map.ini” is setup for this mission:

```
MissionObjectiveList  
    MissionObjectiveTag = SCRIPT:WBTutorialObjectiveText_01  
    ;Chase after the Goblins  
    MissionObjectiveTag = SCRIPT:WBTutorialObjectiveText_02  
    ;Kill the ambush  
    MissionObjectiveTag = SCRIPT:WBTutorialObjectiveText_03  
    ;Find the Goblin base and destroy it  
    BonusMissionObjectiveTag = SCRIPT:WBTutorialBonusObjectiveText_01  
    ;Find the reinforcements
```

END

```
Object GoblinFighterHorde
```

```
    BuildTime = 10
```


```
End
```

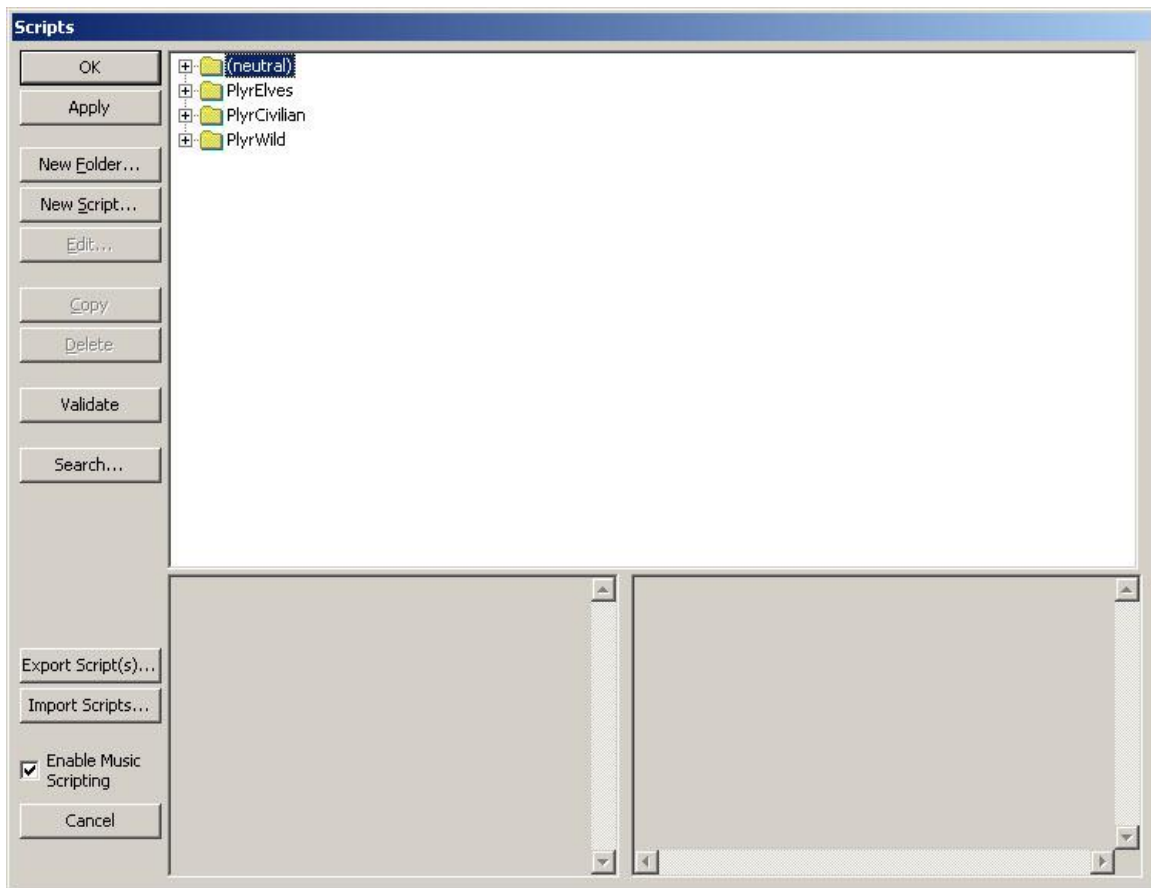
So first we have our MissionObjectiveList. Within this script block are all our mission objectives for our level. To create a mission objective, start with “MissionObjectiveTag” or “BonusMissionObjectiveTag” for bonus objectives, and then reference the “map.str” file for the text associated with the objective. The order of the objectives are important as well when referencing them in World builder. The first objective is 1, then the next is 2,

then 3, etc. This will make more sense when we get to that part of scripting objectives. Also make sure to finish each script block with an “End”. And the final edit we do in this file is to change the build time of a Goblin Fighter Horde. One last thing to note is that in order to comment something out, basically write something that you don’t want the code to actually process, you need to put a “;” in front of it.

Scripting

After your map has been laid out, and your “map.str” and “map.ini” files have been setup it’s time to start scripting. To understand scripting you first need to understand how our

scripting tool works. To open up the scripts menu, click on this icon  near the top of your World Builder window. This will bring up the Script window, which looks like this:

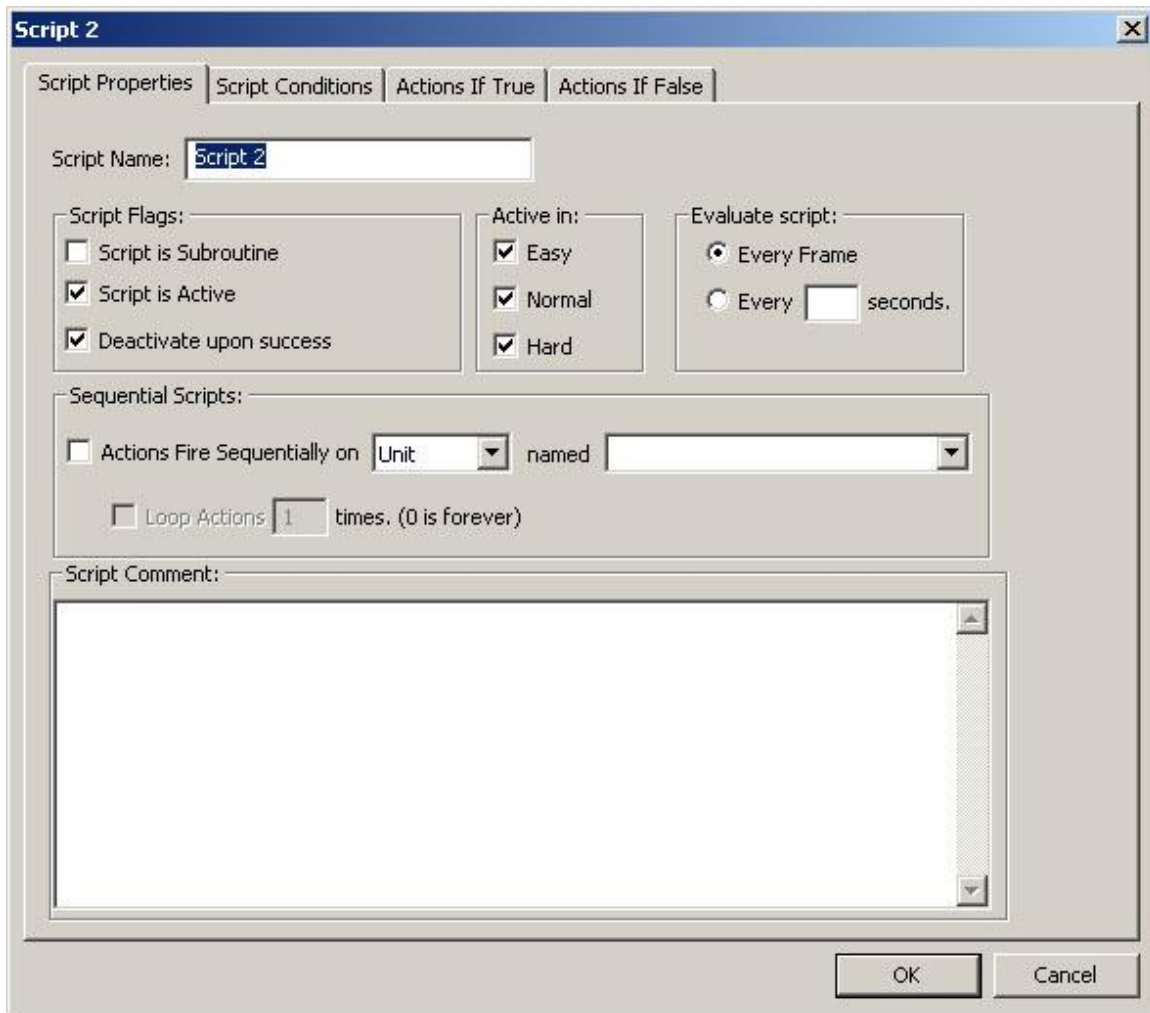


There are a lot of buttons on this screen, so here are brief descriptions of what each one does:

- **OK** – This will save all changes you’ve made to any scripts you’ve edited and close the Scripts window.

- **Apply** – This will save all changes you’ve made, but keep the Scripts window open so you can keep working.
- **New Folder** – This will create a new folder under a Player folder.
- **New Script** – This will bring up a new script window.
- **Edit...** - This will allow you to open any existing script you have selecting. Alternatively you can double left click on any script to edit it as well.
- **Copy** – This will create a copy of any script you have selected and rename the script with a numeric value at the end.
- **Delete** – This will delete any script you have selected.
- **Validate** – This will check all the script in every folder to make sure that their conditions and actions have valid variables. If any script comes up that has a bad one, a “?” will appear on the script alerting you that something is wrong with it.
- **Search...** - This is a rudimentary search tool that allows you to enter a variable and this tool will search through all the scripts looking for that variable. Keep in mind that this tool doesn’t work well when you use very complicated names, so try to keep all your variables short.
- **Export Script(s)...** - This will bring up the Export Scripts window allowing you to export many different things from your map. With this tool, you can copy scripts from one mission to another.
- **Import Scripts...** - This will allow you to import a .scb file, which contains any scripts you have exported recently.
- **Enable Scripting Music** – This allows the engine to automatically take over any music scripting for your mission. If you wish to script your own music and when it changes, then check this box off.
- **Cancel** – This will undo any changes you have made and will close the Scripts window.
- **Top Middle Window** – This window contains all the folders and scripts in your mission. Any Players that exist in your mission will have their own parent folders which can never be deleted. Any scripts or folders within them however can be deleted.
- **Bottom Left Window** – This window will show you any descriptions that a designer may have attached to their scripts.
- **Bottom Right Window** – This window will show you all the script conditions and script actions for any script you currently have selected.

Now that you have an idea of what all the actions you can perform in the Scripts window, let’s take a look at an actual script that you will be editing. Click on the **New Script** button and it will bring up this window.

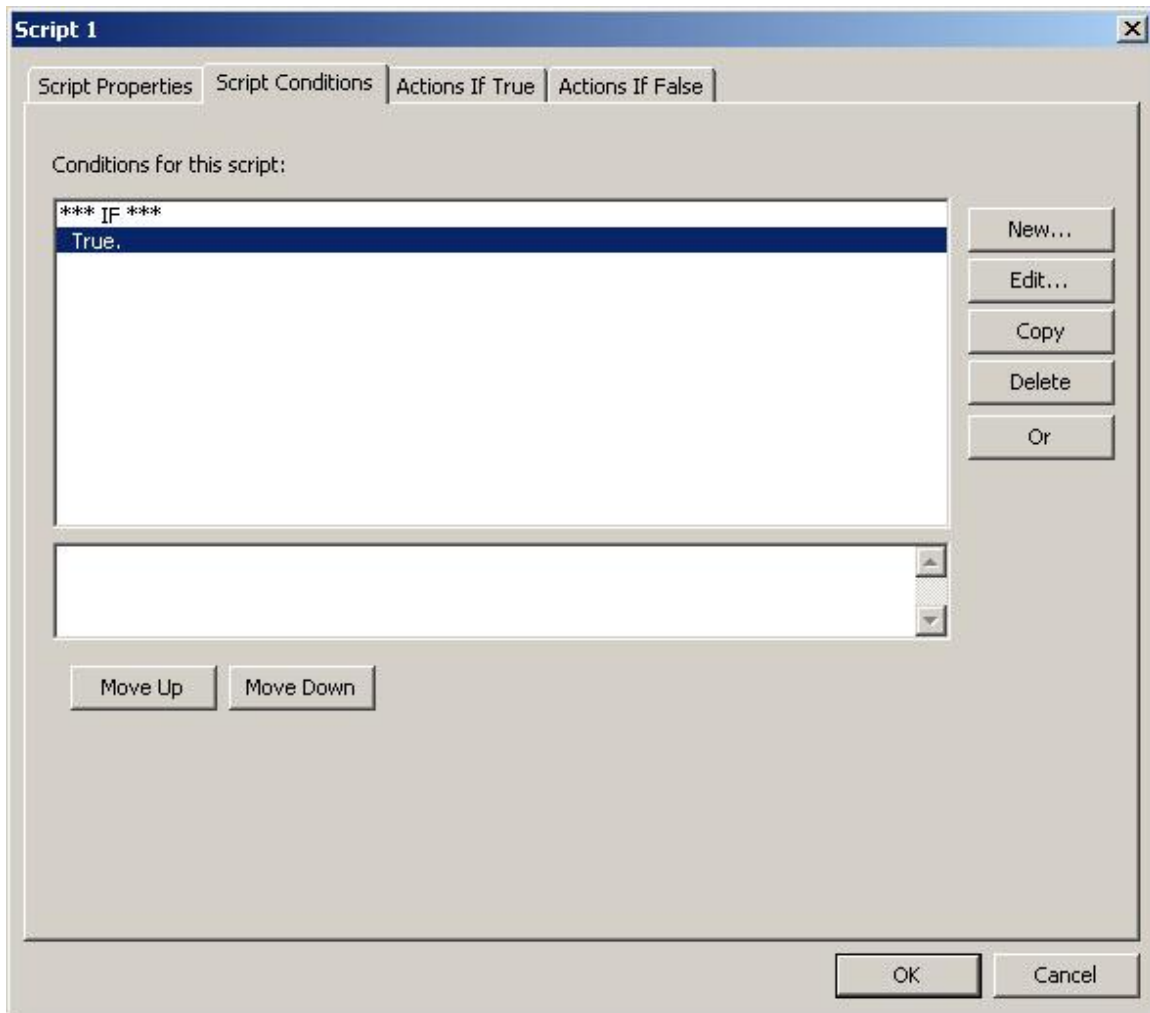


Each script has four tabs at the top that you can cycle through. We'll go over each one in detail. Here's a brief description for all the actions you can do on the **Script Properties** tab:

- **Script Name** – This is where you name your script. You can name it anything you like, but try to keep it simple.
- **Script Flags:**
 - **Script is Subroutine** – Marking a script as a Subroutine means that something has to tell this script to run. In other words, even if all this script's conditions fire true, this script still won't execute unless something tells it to. This is most commonly used when scripting Teams.
 - **Script is Active** – If this is checked, then the script will check it's script conditions to see if they're firing true. If it's unchecked, then this script won't do any checking at all.
 - **Deactivate upon success** – If this is checked, then the script will become inactive, basically like un-checking the Active field. When this field is unchecked, then the script will fire and re-evaluate itself depending on the settings in the **Evaluate Script** portion, and then fire again and again until something else disables it.
- **Active in:**

- **Easy, Medium, Hard** – These are the difficulty settings of the game. For example, if all three of these are checked, then this script will fire in any difficulty setting. If only Hard is checked, then this script will only fire if the Player chooses Hard difficulty when playing the game.
- **Evaluate script:**
 - **Every frame** – This means that the script, when active, will check it's conditions every logic frame (logic frame is about 6 frames to 1 second ratio) and evaluate if all those conditions are true.
 - **Every [] seconds** – This is to allow you to enter a value in seconds for the script to check it's conditions and see if they are true.
- **Sequential Scripts:**
 - **[] Actions fire sequentially** – When this box is checked, the actions listed out on the script will fire sequentially. What this means is that if you have a script that tells a unit to follow a path and then guard for 5 seconds, the script will wait for the first action to complete, the unit following the path, then it will initiate the next action in the script, the unit guarding for 5 seconds. Sequential scripts are extremely useful when you want a unit or a team to do a series of actions in order.
 - **On [Unit\Team]** – This is where you specify which type of group you wish this script to fire sequentially on. Setting this will update the field next to it with the appropriate names.
 - **Named []** – This is where you choose which Unit or Team you wish to have this script fire sequentially on.
 - **[] Loop Actions [] times** – Checking this box will allow you to specify how many times you wish to have the script execute itself before disabling itself. Setting this it's number of times to 0 will make this script execute forever, essentially never disabling itself. This is useful for setting up patrols.
- **Script Comment:** - This is where you can enter in a description of what this script does. Or you can use this to remind you that another script may be enabling this script, etc. Any information you want to remember about this script should go in this field.

The next tab over is the **Script Conditions** and looks like this:

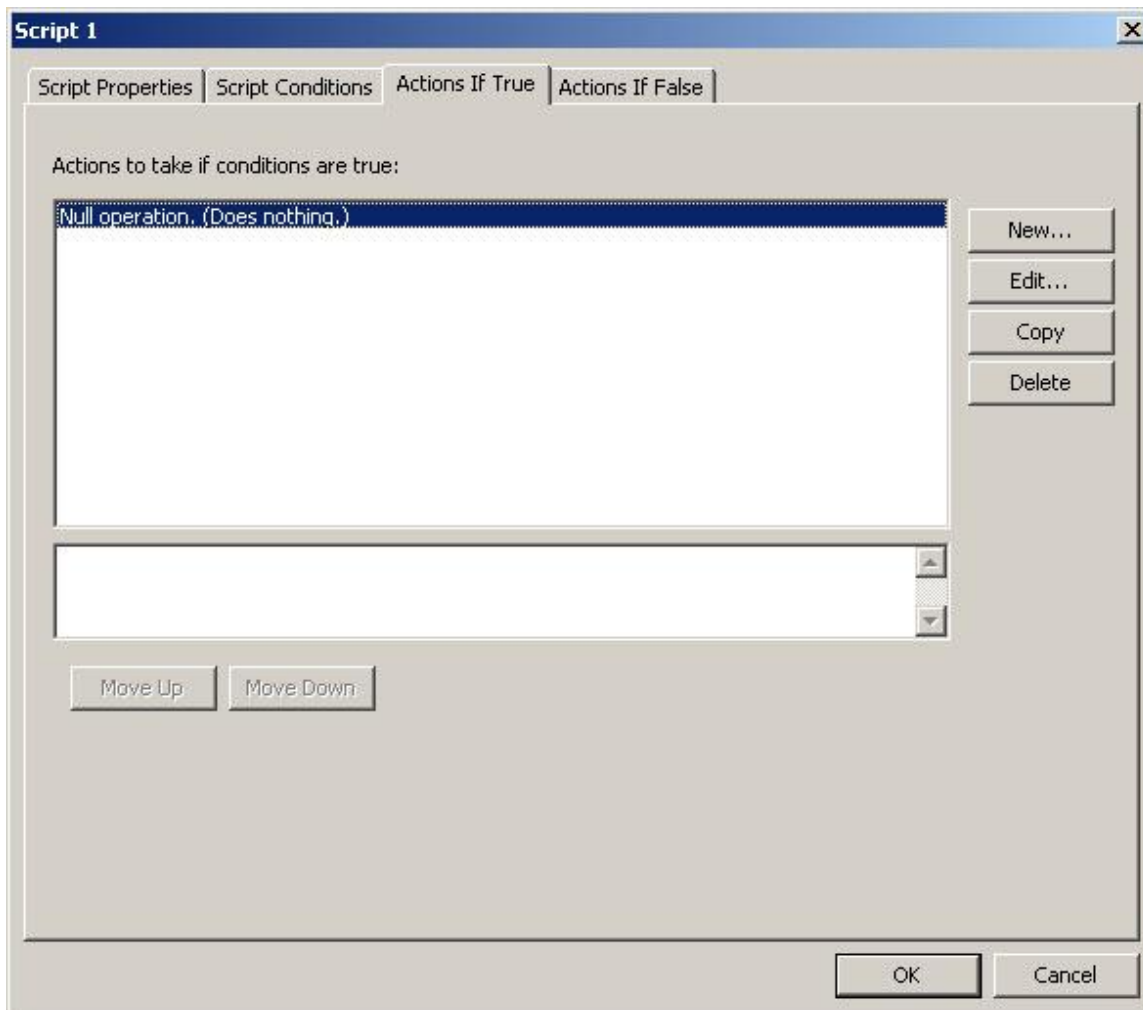


Script conditions are basically what need to happen in order for this script to fire True. When the script fires True, it executes all the actions in the “Actions if True” tab. If the script fires false, basically if the conditions setup in this tab aren’t met, then any actions in the “Actions if False” tab will fire off. Now for a brief description as to what all the buttons do.

- **Conditions for this script:** - This window will show you all the current conditions you have setup.
- **New...** - This will bring up a new window with a list of conditions you are able to choose from.
- **Edit...** - This will allow you to change any script condition you currently have selected. Alternatively, you can double left click any script condition and it will also allow you to edit that condition.
- **Copy** – This will create a copy of any script condition you currently have selected.
- **Delete** – This will delete any script condition you currently have selected.
- **Or** – This will add in an “or” line to your script condition window. So if you want condition 1 to fire OR condition 2 to fire, this is how you set that up.
- **Move Up** – This will allow you to order your script conditions in case you need something to be checked first before other conditions.

- **Move Down** – Exact same thing as Move Up only it will move the conditions down the list.

And finally, the **Actions if True** tab and the **Actions if False** tab are practically identical so we'll just take a look at the **Actions if True** tab:

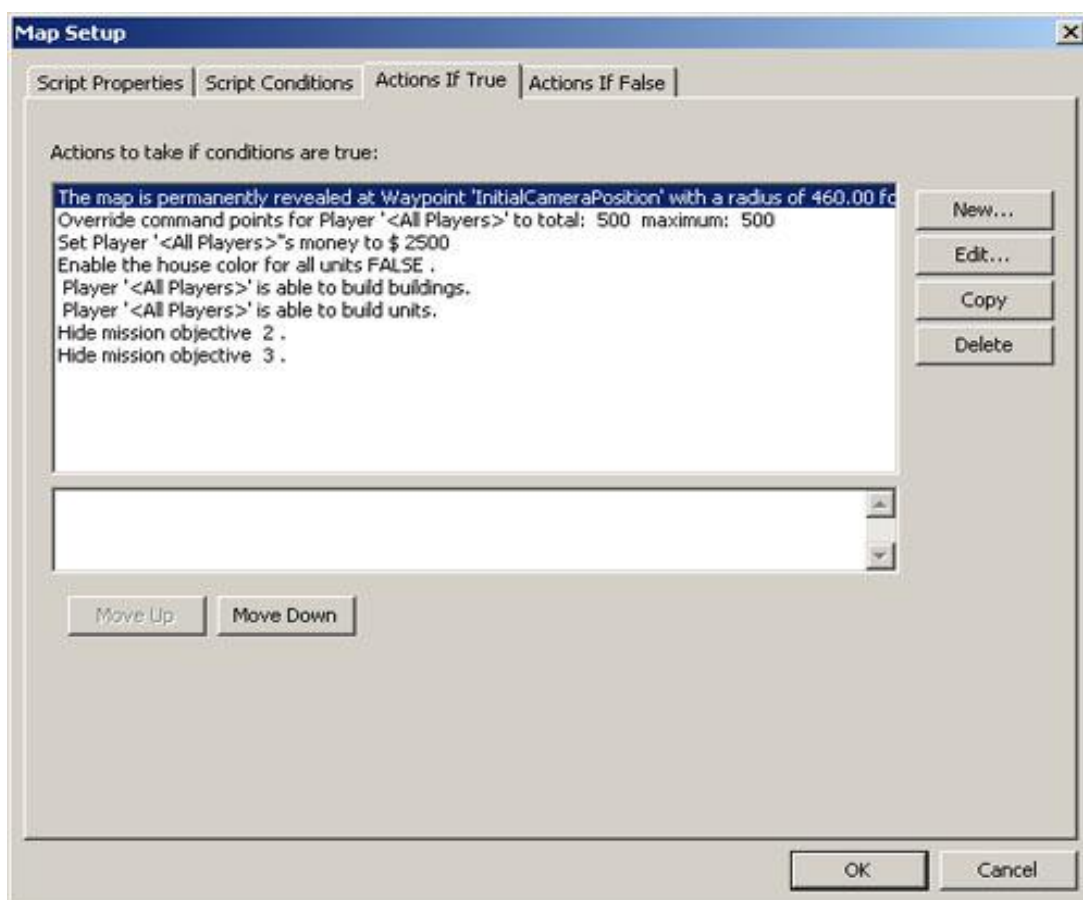


Script actions are what drive your mission. These are what tell units and teams to move around or attack. This will also help build teams and bases up as well as triggering world events like the casting of spells or summoning of units.

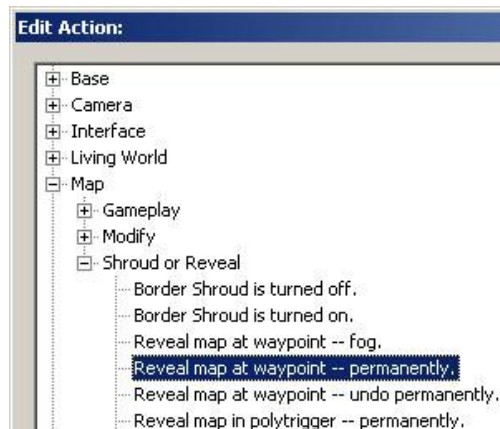
- **Actions to take if conditions are true/false:** - This will display all the actions you have setup for this script.
- **New...** - This will bring up a window for you to choose your actions from.
- **Edit...** - This will allow you to change any script action you currently have selected. Alternatively, you can double left click any script action and it will also allow you to edit that action.
- **Copy** – This will create a copy of any script action you currently have selected.
- **Delete** – This will delete any script action you currently have selected.

- **Move Up** – This will allow you to order your script actions in case you want something acting before something else. This is especially useful when setting up Sequential Scripts.
- **Move Down** – Exact same thing as Move Up only it will move the actions down the list.

Now that you have a reference for what actions you can take within the scripting system, let's setup all the scripts you'll need to run this mission. The first step that you'll want to do is setup all the script actions that need to take place on frame 1. Write a script called "Map Setup". While the name of the script really doesn't matter, its name implies that there are scripts inside setting up the map. Keeping the names of your scripts simple will help others understand what you're trying to achieve with them. Here's the "Map Setup" script written for this map:



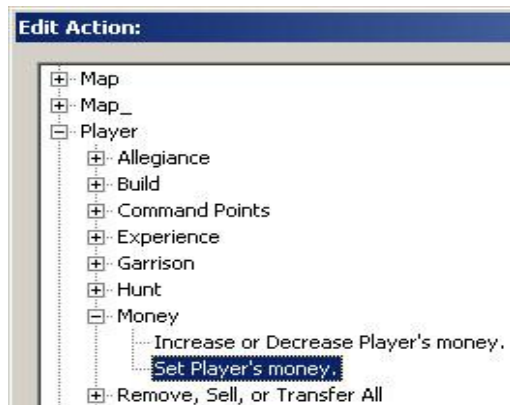
The first action we do is reveal the starting area for the Player, basically where the Player's units are. But this will also show us the first Goblin Warrior Horde. This script action can be found in the picture to the right. I had to reveal it with a radius of 460 in order to get the first Goblin Warrior Horde out of the shroud so the Player could see it in the intro cine.



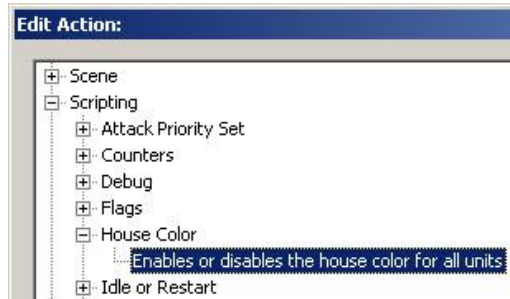
The next action we do is to set all the Player's command points. The way this script is setup not only covers the Human Player's command points, but all AI Player's as well. The action to do this can be found in the picture to the right. When picking which Player I want to set their command points for, I use the generic value, "All Players".



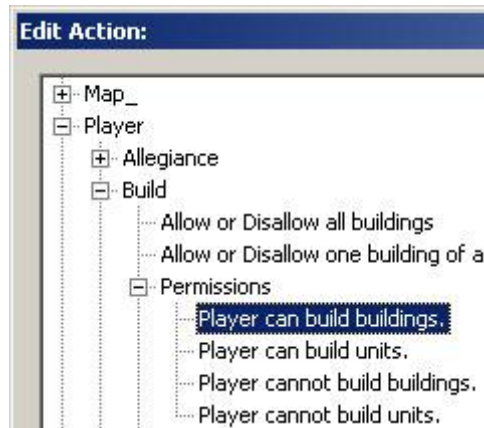
Then we setup all the Players' money. Again I use the generic value, "All Players" because the AI needs money to build it's units and the Human Player will be acquiring a Barracks later on.



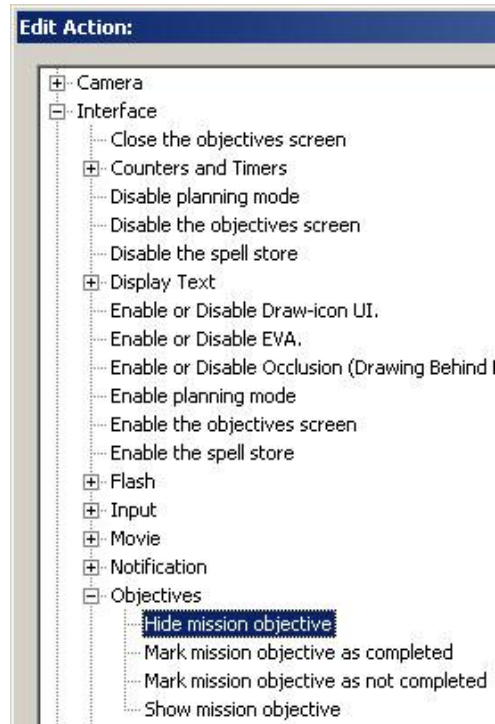
This next script the designers use so that the units themselves have their default color scheme on for all the units in their maps. In multiplayer maps, this color is determined by the Player. It's always on by default so we set it to False so that it doesn't show up. However, this isn't a rule or anything, it's merely a preference.



The next two scripts are very important. Any time you want the AI to build units or structures, you must set these scripts up. By default the AI is never allowed to build anything. And again, using the generic, "All Players" value insures that all the AI Players you have setup on the map will be affected by these script actions. So make sure you always setup the actions, "Player can build buidings." and "Player can build units".



And finally, if you setup your map to have mission objectives, you'll want to make sure to hide the ones you'll want to show up later on in the mission. The reason there is a number associated with them, is because it's the order they are listed in the map.ini file associated with this mission. We'll get to that later on though.



From the above script, you can now get a sense for how the “Edit Action” window is setup. Basically there are parent directories with subdirectories and then the actual actions inside them. For the remainder of the tutorial, and for the sake of space, we will start referring to each action as, “**Action:**” and then the remainder of directories where the script lies. For example, if I was telling you again about overriding a Player’s command points, I would say, “Now create a script action “**Action:** Player\Command Points\Override player command points””. We will also refer to script conditions in the same way, prefacing them with “**Condition:**”.

Intro Cine

All the missions we work on have a brief moment at the beginning called the Intro Cine. This is basically to set the player up with what’s happening in the mission as well as their first objective. For this mission I setup the Intro Cine inside the “(neutral)” player and created its own folder called “Intro Cine”. There are 5 scripts I have created which compose the intro cine:

- **Intro Cine – Part 01**
 - **Condition:** Scripting\True.
 - **Action:** Interface\Input\User input -- disable.
 - **Action:** Camera\Modes\Letterbox mode -- begin.
 - **Action:** Scripting\Timer\Seconds countdown timer -- set.

- **Timer named:** 'Timer - Intro Cine Part 02 Go'
 - **Expire in** 2.00 seconds
- **Intro Cine – Part 02**
 - **Condition:** Scripting\Timer expired.
 - **Timer named:** 'Timer - Intro Cine Part 02 Go'
 - **Action:** Interface\Display Text\Show text message to user.
 - **Localized String:** 'SCRIPT:WBTutorialIntroText_01'
 - **For** 5.00 seconds
 - **Action:** Scripting\Timer\Seconds countdown timer -- set.
 - **Timer named:** 'Timer - Intro Cine Part 03 Go'
 - **Expire in** 5.00 seconds
- **Intro Cine – Part 03**
 - **Condition:** Scripting\Timer expired.
 - **Timer named:** 'Timer - Intro Cine Part 03 Go'
 - **Action:** Camera\Move\Move the camera to a location.
 - **Move camera to:** 'WP - Look at Goblin Retreater'
 - **In:** 3.00 seconds
 - **Camera shutter:** 0.00 seconds
 - **Ease-in:** 1.00 seconds
 - **Ease-out:** 1.00 seconds
 - **Action:** Scripting\Timer\Seconds countdown timer -- set.
 - **Timer named:** 'Timer - Intro Cine Part 04 Go'
 - **Expire in** 5.00 seconds
 - **Action:** Interface\Display Text\Show text message to user.
 - **Localized String:** 'SCRIPT:WBTutorialIntroText_02'
 - **For** 3.00 seconds
- **Intro Cine – Part 04**
 - **Condition:** Scripting\Timer expired.
 - **Timer named:** 'Timer - Intro Cine Part 04 Go'
 - **Action:** Unit (includes Structures) -- Action\Move\Unit move to a location.
 - **Unit named:** 'Goblin Retreater'
 - **Waypoint:** 'WP - Goblin Retreater Run Here'
 - **Action:** Interface\Display Text\Show text message to user.
 - **Localized String:** 'SCRIPT:WBTutorialIntroText_03'
 - **For** 5.00 seconds
 - **Action:** Scripting\Timer\Seconds countdown timer -- set.
 - **Timer named:** 'Timer - Intro Cine Part 05 Go'
 - **Expire in** 5.00 seconds
- **Intro Cine – Part 05**
 - **Condition:** Scripting\Timer expired.
 - **Timer named:** 'Timer - Intro Cine Part 05 Go'
 - **Action:** Camera\Move\Reset to the default view.
 - **Waypoint:** 'InitialCameraPosition'
 - **Taking:** 0.00 seconds
 - **Ease-in:** 0.00 seconds
 - **Ease-out:** 0.00 seconds
 - **Action:** Camera\Modes\Letterbox mode -- end.

- **Action:** Interface\Input\User input -- enable.
- **Action:** Interface\Notification\Display popup notification box.
 - **Type:** 'NewObjective'
 - **Localized String:** 'SCRIPT:WBTutorialObjectiveText_01'
 - **For:** 5.00 seconds

Objectives

Under the PlyrElves folder, I have setup how all the objectives are going to be handled. Basically what determines them being complete as well as showing the new objectives as the mission progresses. Create a folder in this player folder called "Objectives". Here are the following scripts associated with the objectives for this mission:

- **Objective 01 Complete**
 - **Condition:** Player\Area\Player has units in an area
 - **Player:** <Local Player>
 - **Area:** 'AT - Ambush Team Attack'
 - **Action:** Interface\Objectives\Mark mission objective as completed
 - **Objective:** 1
 - **Action:** Interface\Objectives\Show mission objective
 - **Objective:** 2
 - **Action:** Interface\Notification\Display popup notification box.
 - **Type:** 'New Objective'
 - **Localized String:** 'SCRIPT:WBTutorialObjectiveText_02'
 - **For:** 5 seconds
- **Objective 02 Complete**
 - **Condition:** Unit (includes Structures)\Attacked, Destroyed or Exists\Unit is destroyed.
 - **Unit:** 'Goblin Retreater'
 - ***AND* Condition:** Scripting\Flag compared to a value
 - **NOTE:** Anytime you have two consecutive script conditions that aren't separated by an "OR", an "*AND*" will appear next to any condition after the first.
 - **Flag named:** 'PlyrWild/Flag - Ambush Team Destroyed'
 - **NOTE:** This flag will not show up in the list until it is created in a later script called "**Goblin Ambush Team Destroyed**". The reason a flag is called is explained where that script is described in this tutorial.
 - **IS:** TRUE
 - **Action:** Interface\Objectives\Mark mission objective as completed
 - **Objective:** 2
 - **Action:** Interface\Objectives\Show mission objective
 - **Objective:** 3
 - **Action:** Interface\Notification\Display popup notification box.
 - **Type:** 'New Objective'
 - **Localized String:** 'SCRIPT:WBTutorialHintText_02'
 - **For:** 5 seconds

- **Objective 03 Complete**
 - **Condition:** Player\Area\Player doesn't have units in an area.
 - **Player:** 'PlyrWild'
 - **Area:** 'AT - Goblin Base'
 - **Action:** Interface\Objectives\Mark mission objective as completed
 - **Objective:** 3
 - **Action:** Scripting\Flags\Set flag to value.
 - **Flag named:** 'Flag - Mission won'
 - **To:** TRUE
- **Bonus Objective 1 Complete**
 - **Condition:** Player\Area\Player has units in an area
 - **Player:** <Local Player>
 - **Area:** 'AT - Transfer Reinforcement Elves'
 - **Action:** Interface\Objectives\Mark mission objective as completed
 - **Objective:** 4

Mission Win & Loss

Also under the PlyrElves folder I have the **Mission Won** and **Mission Lost** scripts. They're scripted as follows:

- **Mission Won**
 - **Condition:** Scripting\Flag compared to a value
 - **Flag named:** 'PlyrWild/Flag - Ambush Team Destroyed'
 - **NOTE:** This Flag was setup in the **Objective 03 Complete** script.
 - **Action:** Win & Loss\Announce victory.
- **Mission Lost**
 - **Condition:** Team\Attacked, Destroyed, or Exists\Team is destroyed.
 - **Team:** 'teamPlyrElves'
 - **NOTE:** Designating 'teamPlyrElves' will cover all units owned by the local player as all the locals players units default to that team.

PlyrElves Scripts

The remaining script under the PlyrElves folder is to alert the Player that they're making headway towards their final goal. It's scripted as follows:

- **Found Patrol**
 - **Condition:** Player\Sighted\Player has sighted Team.
 - **Team:** 'PlyrWild/Goblin Patrol Team 01'
 - **Discovered by:** '<Local Player>'
 - *****OR*** Condition:** Player\Sighted\Player has sighted Team.
 - **Team:** 'PlyrWild/Goblin Patrol Team 02'
 - **Discovered by:** '<Local Player>'

- **Action:** Interface\Notification\Display popup notification box.
 - **Type:** 'Hint'
 - **Localized String:** 'SCRIPT:WBTutorialHintText_03'
 - **For:** 5 seconds

PlyrCivilian Scripts

Because the Elven Reinforcements start on the Civilian Player, I have placed the script that transfers them from the Civilian Player to the Elves Player into the PlyrCivilian folder. The script that handles this is as follows:

- **Elven Reinforcements**
 - **Condition:** Player\Area\Player has units in an area
 - **Player:** <Local Player>
 - **Area:** 'AT - Transfer Reinforcement Elves'
 - **Action:** Unit (includes structures) -- Other\Transfer a specific unit to the control of Player.
 - **Unit:** 'Elf Barracks'
 - **Command of Player:** '<Local Player>'
 - **Action:** Unit (includes structures) -- Other\Transfer a specific unit to the control of Player.
 - **Unit:** 'Elf Reinforcement 01'
 - **Command of Player:** '<Local Player>'
 - **Action:** Unit (includes structures) -- Other\Transfer a specific unit to the control of Player.
 - **Unit:** 'Elf Reinforcement 02'
 - **Command of Player:** '<Local Player>'
 - **Action:** Interface\Notification\Display popup notification box.
 - **Type:** 'BonusObjectiveCompleted'
 - **Localized String:** 'SCRIPT:WBTutorialHintText_03'
 - **For:** 5 seconds

Goblin Base

Within the PlyrWild folder lies the Goblin Base team building scripts. Up until now, all the scripts you've created haven't required any of their Script Properties to be edited. For all the following scripts, you will have to make sure to mark them as **Subroutines**. Create a folder called "Goblin Base" and here are the scripts within that folder that handle the team building for the base:

- **Build Defense Teams** - Attach this script to the **Production\Condition** field of the **Goblin Base Defense Team's** Team Template.
 - Under the Script Properties tab (first tab on a new script) the following check boxes should be marked as follows:
 - **Script is a Subroutine:** Checked.

- **Deactivate upon success:** Not checked.
 - **NOTE:** The reason that this script doesn't have to deactivate upon success is because this script will only ever run once per instance of a team created. Deactivating it may cause future instances of the same team to not be created.
 - **Condition:** Scripting\Counter compared to a value.
 - **Counter:** 'Counter - Goblin Patrols Dead'
 - **IS:** Greater Than or Equal To
 - **Value:** 3
 - **Action:** Scripting\Debug\Null operation.
 - **NOTE:** The reason the action is a Null Operation is because this script is attached to a team that is only checking for the condition to fire true. Once that condition fires true, then the action is to build the team, which is handled by the Team Template.
- **Build Attack Teams** - Attach this script to the **Production\Condition** field of the **Goblin Base Attack Team's** Team Template.
 - Under the Script Properties tab (first tab on a new script) the following check boxes should be marked as follows:
 - **Script is a Subroutine:** Checked.
 - **Deactivate upon success:** Not checked.
 - **Condition:** Scripting\Counter compared to a value.
 - **Counter:** 'Counter - Goblin Patrols Dead'
 - **IS:** Greater Than or Equal To
 - **Value:** 3
 - *****OR*** Condition:** Player\Area\Player has units in an area
 - **Player:** <Local Player>
 - **Area:** 'AT - Build Goblin Defense and Attack Teams'
 - **Action:** Scripting\Debug\Null operation.
- **Defense Team Built** Attach this script to the **Behavior Script Triggers\On Create** field of the **Goblin Base Defense Team's** Team Template.
 - Under the Script Properties tab (first tab on a new script) the following check box should be marked as follows:
 - **Script is a Subroutine:** Checked.
 - **Condition:** Scripting\True.
 - **NOTE:** The reason the condition is just True is because this script is attached to a section of the Team Template window that act as conditions themselves. Therefore, the only parameters this script needs is actions.
 - **Action:** Team -- Action\Guard\Guard -- area.
 - **Team:** '<This Team>'
 - **Area:** 'AT - Goblin Base'
- **Attack Team Built** Attach this script to the **Behavior Script Triggers\On Create** field of the **Goblin Base Attack Team's** Team Template.
 - Under the Script Properties tab (first tab on a new script) the following check box should be marked as follows:
 - **Script is a Subroutine:** Checked.
 - **Condition:** Scripting\True.

- **Action:** Team -- Action\Attack or Hunt\Team attacks Team.
 - **Team:** '<This Team>'
 - **Attack on Team:** 'PlyrElves/teamPlyrElves'

Goblin Patrols

This next folder handles the four teams that patrol the roads. This next set of scripts also introduces us to a new parameter, Sequential Scripts. Name the folder "Goblin Patrols" and here are the scripts that handle each of the teams:

- **Goblin Patrol Team Destroyed** - Attach this script to the **Behavior Script Triggers\On Destroyed [100]** field of each of the **Goblin Patrol's** (01-04) Team Templates.
 - **Condition:** Scripting\True
 - **Action:** Scripting\Counters\Increment counter.
 - **Add:** 1
 - **Counter named:** 'Counter - Goblin Patrols Dead'
- **Goblin Patrol 01 - Patrol**
 - Under the Script Properties tab (first tab on a new script) the following check box and parameters should be marked as follows:
 - **Actions Fire Sequentially:** Checked.
 - **On:** [Team]
 - **Named:** [Goblin Patrol Team 01]
 - **Loop Actions [0] times.:** Checked
 - **NOTE:** We're setting the Loop Actions to 0 so that this team will never stop patrolling. Once it gets to the end of it's script actions it will repeat it forever.
 - **Condition:** Scripting\True
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 01'
 - **Waypoint Path:** 'WP Path - Patrol Path 01'
 - **As a team is:** FALSE
 - **Information is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 01'
 - **Guards for:** 5 seconds
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 01'
 - **Waypoint Path:** 'WP Path - Patrol Path 02'
 - **As a team is:** FALSE
 - **Information is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 01'
 - **Guards for:** 5 seconds

- **Goblin Patrol 02 - Patrol**
 - Under the Script Properties tab (first tab on a new script) the following check box and parameters should be marked as follows:
 - **Actions Fire Sequentially:** Checked.
 - **On:** [Team]
 - **Named:** [Goblin Patrol Team 02]
 - **Loop Actions [0] times.:** Checked
 - **Condition:** Scripting\True
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 02'
 - **Waypoint Path:** 'WP Path - Patrol Path 02'
 - **As a team is:** FALSE
 - **In formation is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 02'
 - **Guards for:** 5 seconds
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 02'
 - **Waypoint Path:** 'WP Path - Patrol Path 01'
 - **As a team is:** FALSE
 - **In formation is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 02'
 - **Guards for:** 5 seconds
- **Goblin Patrol 03 - Patrol**
 - Under the Script Properties tab (first tab on a new script) the following check box and parameters should be marked as follows:
 - **Actions Fire Sequentially:** Checked.
 - **On:** [Team]
 - **Named:** [Goblin Patrol Team 03]
 - **Loop Actions [0] times.:** Checked
 - **Condition:** Scripting\True
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 03'
 - **Waypoint Path:** 'WP Path - Patrol Path 03'
 - **As a team is:** FALSE
 - **In formation is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 03'
 - **Guards for:** 5 seconds
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 03'
 - **Waypoint Path:** 'WP Path - Patrol Path 04'
 - **As a team is:** FALSE
 - **In formation is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 03'
 - **Guards for:** 5 seconds

- **Goblin Patrol 04 - Patrol**
 - Under the Script Properties tab (first tab on a new script) the following check box and parameters should be marked as follows:
 - **Actions Fire Sequentially:** Checked.
 - **On:** [Team]
 - **Named:** [Goblin Patrol Team 04]
 - **Loop Actions [0] times.:** Checked
 - **Condition:** Scripting\True
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 04'
 - **Waypoint Path:** 'WP Path - Patrol Path 04'
 - **As a team is:** FALSE
 - **In formation is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 04'
 - **Guards for:** 5 seconds
 - **Action:** Team -- Action\AttackMove\Follow a waypoint path.
 - **Team:** 'Goblin Patrol Team 04'
 - **Waypoint Path:** 'WP Path - Patrol Path 03'
 - **As a team is:** FALSE
 - **In formation is:** FALSE
 - **Action:** Team -- Other\SequentialScript\Guard -- number of seconds.
 - **Team:** 'Goblin Patrol Team 04'
 - **Guards for:** 5 seconds

PlyrWild Scripts

The last three scripts handle transferring the Goblin Retreater and the Goblin Ambush. The scripts are setup as follows:

- **Transfer Goblin Retreater to this Player**
 - **Condition:** Unit (includes Structures)\Area or Path\Unit inside an area.
 - **Unit:** 'Goblin Retreater'
 - **Area:** 'AT - Transfer Goblin Retreater to Wild Player'
 - **Action:** Unit (includes Structures) -- Other\Allegiance\Transfer a specific unit to the control of Player.
 - **Unit:** 'Goblin Retreater'
 - **Player:** 'PlyrWild'
- **Goblin Ambush Team Destroyed**
 - **Condition:** Team\Attack, Destroyed, or Exists\Team is destroyed.
 - **Team:** 'Ambush Team'
 - **Action:** Scripting\Flags\Set flag to value.
 - **Flag named:** 'Flag - Ambush Team Destroyed'
 - **To:** TRUE
 - **NOTE:** There's a very good reason that we set a Flag when a team is destroyed and that's because the engine only remembers the team being destroyed on the frame it was

destroyed. So if a team dies, and 5 frames later you check to see if that team was destroyed, your condition won't fire true. If you need to check if a team is destroyed, make sure you setup a script like this that evaluates every single frame and then have it set a Flag. Flags will always remain in memory, therefore they can be checked at any time.

- **Ambushers Attack**
 - **Condition:** Player\Area\Player has units in an area
 - **Player:** <Local Player>
 - **Area:** 'AT - Ambush Team Attack'
 - **Action:** Team -- Other\Settings\Set the general attitude of Team.
 - **Team:** 'Ambush Team'
 - **Attitude to:** Aggressive
 - **Action:** Team -- Action\Attack or Hunt\Team attacks Team.
 - **Team:** 'Ambush Team'
 - **Attack on Team:** 'PlyrElves/teamPlyrElves'

And that's it for the scripting tutorial. I hope in the least you've familiarized yourself with our scripting tool, and go on to explore all the other scripts that reside within it's database. And on that note, good luck and happy scripting!