

**Tomb Raider NextGen**  
**© Core Design Ltd**

**Subject:** AI Production line  
**Date:** 2002-09-20  
**Revision:** 1.00

<mailto:stephane@core-design.com>

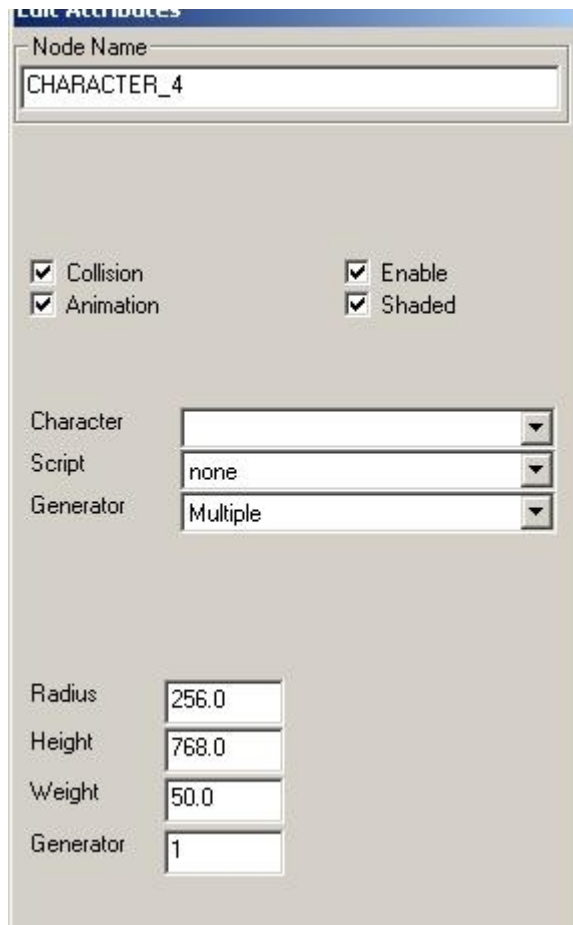
Characters are placed from **WorldEdit**.

First Select the room the room where the character need to placed (with Left Shift) The room selected becomes **red – wired**



Select **Create Character**. A white Cylinder will be displayed. Move it into the correct location. You can rotate the locator (use the **green** arrow as for character direction)

Press Left Ctrl A  
to edit properties of the current character.



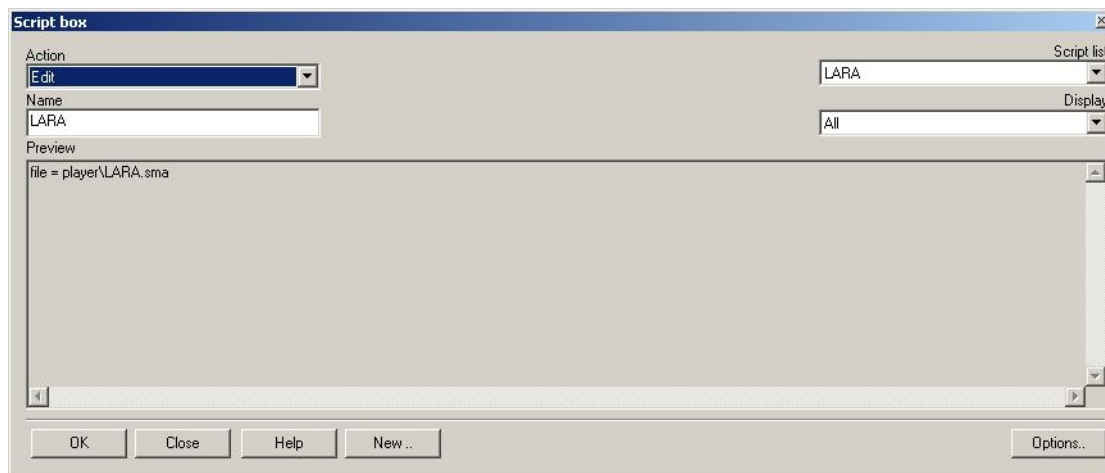
Select the character model from the list (ie : Lara Croft)

You probably needs to create a script for the character. To do that, go to '**Tools**' et Select **Scripts**. (See the Script Box section)

You need to attach a script to the character.

- Only matching scripts (correct character from script and from the mesh model) are allowed (ie, only LARA can get LARA scripts).
- One script can be used by several character.
- An actor is identified by it's name.

## The Script Box



### Script list :

Display list of the script in this level

### Action:

Action button (press Ok to validate)

- Edit : the current script
- Compile : Compile the script. If success, nothing is displayed else an output dialogue appears with the errors. Use EDIT to fix the errors
- Delete : Delete script from the database.
- Edit Symbolic Link: If the script preview display link = <something> or file =<something> you will be able to Rename the link
- Rename: Rename current script.

### Preview:

Display script source code.

Scripts can be external file. So they are displayed as

```
file = script_file.SMA
```

To create a new script, click on New. There is several kind of script

- Blank Script : start from a empty script
- Level Based script: Select from a list a level script. A level script has usual 3 entrypoints : Before execution of the level, During the level and at the end of the level.
- Dialogue script: This is the script for most dialogue , cut scene etc..
- Character: Select the default script for a character.
- File: Select a script from an external file

## Scripts file

Character script is a text file with some command for the character. A script is defined by a set of functions which looks like this

```
public OnInit(objref)
{
    // Comments.
}
```

There are 2 important functions for the character

OnInit  
OnUpdate

OnInit defines the script at initialisation. Usually this is to set up the default state of the character.

Each character can have different states

ACTOR_IDLE	Character is not activated or displayed
ACTOR_ACTIVE	Character is activated
ACTOR_DEAD	Character has been killed

If you look at the OnUpdate() code, you will see that the script will handle the different states like this:

```
public OnUpdate(objref)
{
    if (ACTOR_ACTIVE == state)
    {
        OnThinkActive(actor);
    }
    else
    if (ACTOR_DEAD == state)
    {
        OnThinkDead(actor);
    }
    else
    {
        OnThinkIdle(actor);
    }
    return 0;
}
```

So 3 functions are defined **OnThinkActive()**, **OnThinkDead()** and **OnThinkIdle()**

Let's go back to the init function,

```

public OnInit(objref)
{
    // Set actor state
    SetSuperState(actor, ACTOR_ACTIVE);

    // Set initial animation
    SetAnimation(objref, COPA_LOOKAROUND, 0);

    // Set 'active' state
    SetState(actor, STATE_STARTWANDER);
}

```

The active state is the 'sub' state of the active mode. This is actually set to STARTWANDER. Each character has different possibility of behavior

Example : COP\_A (which is the resource name of Policeman A) has these following states :

STATE_WALKPATROL	Process the patrol. Walk through the waypoint
STATE_STARTPATROL	Define the waypoint the process
STATE_LOOKAROUND	Actor stay and look around
STATE_SPOT	Actor spotted the player
STATE_CHASING	Actor is chasing the player
STATE_WALKPATH	Actor is processing a path to the player
STATE_FINDPATH	Actor is finding a path to the player
STATE_SEEKPLAYER	Actor try to find the player in his field of view
STATE_ATTACK	Actor is attacking the player
STATE_SHOOT	Actor is shooting at the player
STATE_OPENDOOR	Actor is open the door (the Courtyard 's cop)
STATE_WANDER	Actor start from a waypoint and pickup an other to his closest field of view
STATE_STARTWANDER	Actor is processing a 'wandered' path.. He will not attack you.

The List of the animation names (like COPA\_LOOKAROUND) is available from **Animator Editor**

## Create Specific waypoints path

If you want that buddy process a specific path.

First define a function with the list of the waypoints to process

```
OnBuildPathCop5(actor)
{
    BeginPath(actor,1);
    AddWaypoint(actor, streets__street6__cop5_wpoint_0);
    AddWaypoint(actor, streets__street51__cop5_wpoint_18);
    EndPath(actor);
}
```

The name of the waypoint is displayed in worldEdit.

Edit the **OnCustomPatrol** function already provided

```
OnCustomPatrol(actor)
{
    if (__cop5 == actor)
        OnBuildPathCop5(actor);
    SetState(actor, STATE_WALKPATROL);
}
```

The function checks that the current actor is ‘\_\_cop5’ and then call the function `OnBuildPathCop5()`. The second line set the state to `STATE_WALKPATROL`

Since the waypoints are specific per level, you must add a directive to declare the list of the waypoints, characters and other triggers.

To do so, add once

```
#include <level/z1_ms.h>
```

assuming that `z1_ms` is the target named (ie: the name at export)

## Character Triggers

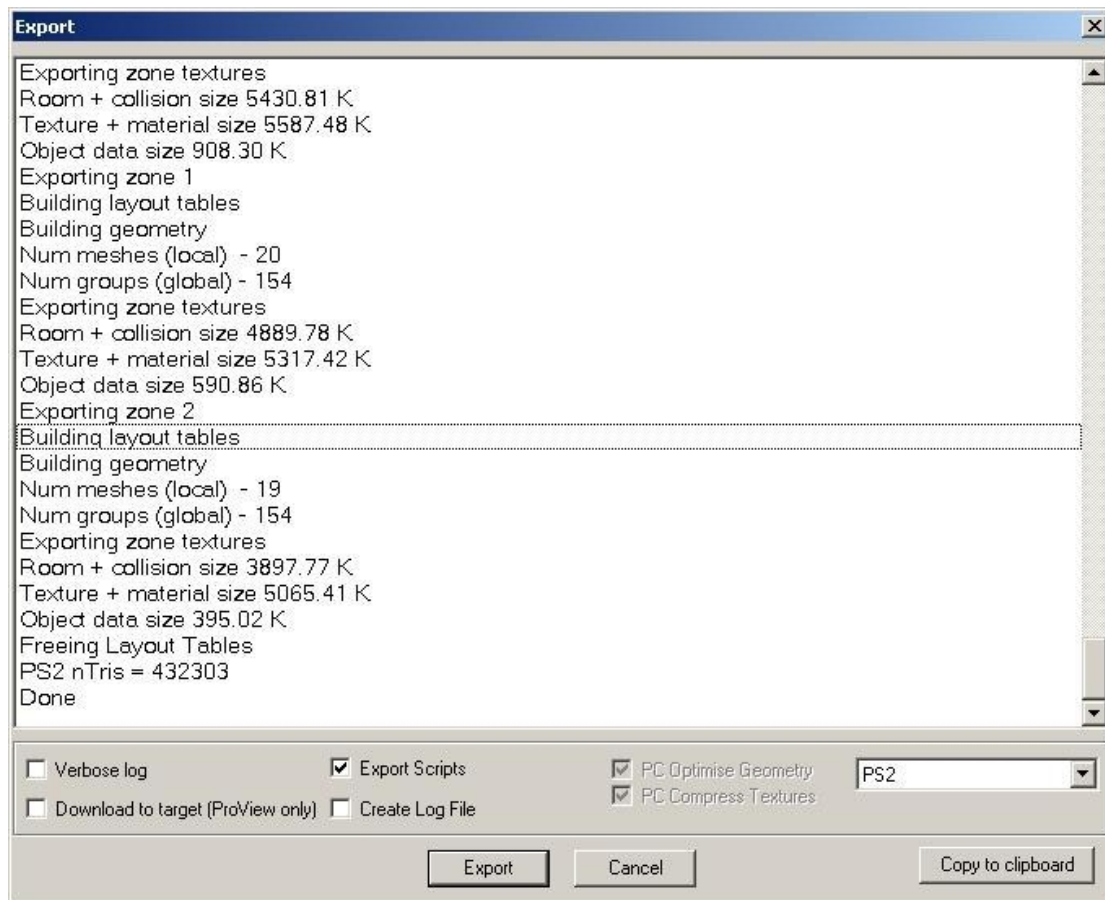
You can create some triggers to activate or deactivate some characters.

1. Create a trigger
2. Resize it into a correct size
3. Press Left Ctrl A
4. In the Trigger Panel, click on 'Active'
5. Press Edit Param List
6. Press Search and select your character
7. In user Param choose 0 for de-activate, 1 for activate.
8. Press Create a new Entry

Note : The characters named must start by a double underscore. If it is not the case, save the project, exit worldedit, and reload the project.

## Exporting the level

When you change the script, you need to export the level again by using the X key and checking the 'export script' button



You must also copy manually the character resource from

Z:\NEXTGEN\GAMECHARS\EXPORT.PS2

to

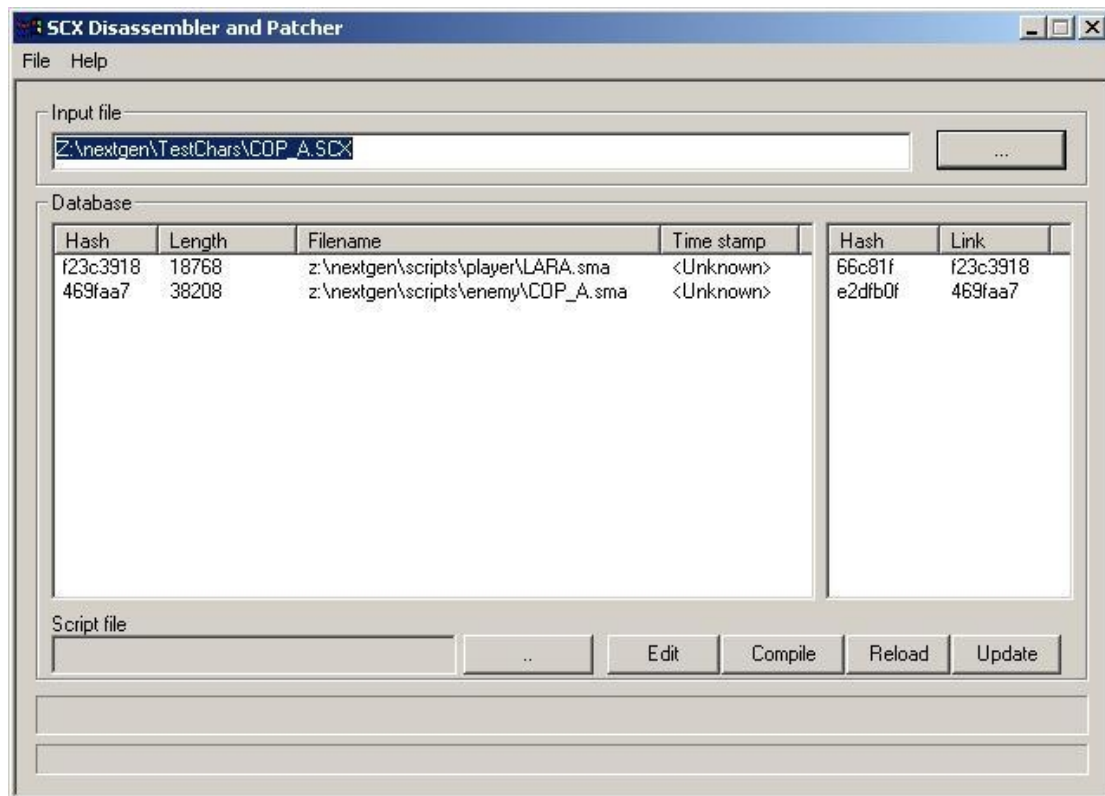
C:\PROGRAM FILES\TOMB RAIDER\COMMON FILES\DATA\CHAR



## SCX files

When exporting a level, you can modify scripts outside WorldEdit.  
To do so, double click on the SCX files which are located in the same path than you RMX files. in order to get the SCX extension, double click on

z:\nextgen\scripts\compiler\install.reg



Press Edit to edit the current script.

Press Compile to compile the current script. If success, you must get the <Updated> in Time stamp column

After updated the scripts, you can restart your level on your Proview (press SELECT)