



# Mod Tools – Object Creation Tutorial

## Table of Contents

<u>1</u>	<u>Introduction</u> .....	3
<u>2</u>	<u>Mod Tools Overview</u> .....	4
<u>3</u>	<u>Create Your Object in 3D Studio Max</u> .....	5
<u>3.1</u>	<u>Sample Object File</u> .....	5
<u>3.2</u>	<u>Objects for the Shed</u> .....	5
<u>3.3</u>	<u>Export the Object from 3D Studio Max</u> .....	7
<u>4</u>	<u>Edit the Shed in OED</u> .....	8
<u>4.1</u>	<u>Import the 3D Studio Max File</u> .....	8
<u>4.2</u>	<u>Edit the Object</u> .....	8
	<u>OED Basics</u> .....	8
	<u>Adding a Fan</u> .....	9
<u>4.3</u>	<u>Export for Mission Editing</u> .....	9
<u>5</u>	<u>Add Object to Item Definitions</u> .....	10
<u>6</u>	<u>Insert Object into a Mission</u> .....	11
<u>7</u>	<u>Pack the Mod</u> .....	12
<u>8</u>	<u>Run the Mod</u> .....	12

# 1 Introduction

With the new mod tools you can add new objects to Delta Force - Black Hawk Down.

This tutorial walks you through the steps for adding a new building. We'll start by reviewing a shed that was created in 3ds max® (3D Studio Max), and then describe the various objects and object names that were added to it. Then you'll walk through the steps of adding game characteristics to it, and inserting it into a Black Hawk Down mission.

While we'll talk about 3D Studio Max in this tutorial, you're the artist, the object designer. You'll need to be familiar with 3D Studio Max already.

Before you start making changes to the game, be sure to make a backup copy of both your Black Hawk Down game directory and your `ModTools` directory (wherever you installed the mod tool files). If you make a mistake while creating your mod files, you may damage some game files.

The mod tools released for Black Hawk Down are not supported. Visit NovaWorld ([www.novaworld.com](http://www.novaworld.com)) and join the forum to discuss them with other gamers.

## 2 Mod Tools Overview

You need to follow a sequence of steps to create a new mod object for Black Hawk Down.

This summary will give you an idea of what's required. You can refer back to this list after you've gone through the tutorial. Later in the document, each step is explained in more detail.

- 1) Create your object in 3D Studio Max
  - a. Draw & name the different parts of your object so they will work in the game
  - b. Export the object into an .ASE file format (save it in your working directory)
- 2) Edit the object in the NovaLogic Object Editor (OED)
  - a. Import the .ASE file into the OED
  - b. Edit the object in OED, adding specific attributes to your object
  - c. Export the OED object to create the .3DA file
- 3) Add the object to the Item Definition file
- 4) Pack the object into your mod file (e.g. `MyMod.PFF`)
- 5) Insert the object to a Black Hawk Down mission using the Mission Editor (MED)
- 6) Pack the mission into your mod file
- 7) Run your mod and play!

**Tip:** To run the Mission Editor with new objects in your mod, run this command from the game directory, or create a shortcut with these parameters:  
`dfbhdmed /mod MyMod.`

**Tip:** To run Black Hawk Down with your new objects and missions, run this command from the game directory, or create a shortcut:  
`dfbhd /mod MyMod.`

## 3 Create Your Object in 3D Studio Max

All the textures and shapes for the sample building, a shed, were created in 3D Studio Max. After you export the file from 3D Studio Max, you can add other game-specific features with the NovaLogic Object Editor (OED).

This document will walk you through the process for creating a game object in 3D Studio Max, focusing on the types of objects required by the Black Hawk Down game engine. As an object designer, you need to be familiar with 3D Studio Max® already.

**Warning:** You need to be running version 3.1 or higher of 3DStudioMax to create new objects for Delta Force - Black Hawk Down.

### 3.1 Sample Object File

A 3D Studio Max file, `ShedH.ASE`, comes with the mod tools release for Black Hawk Down. It shows the new building, a shed, that we'll be using in this tutorial.

The 'H' in the Shed filename stands for High resolution. Lower resolution files aren't provided with this tutorial, but when you create the image for your object, you will need to create resolutions for all four sizes: High, Medium, Small, and Tiny. These different resolutions of the object will be drawn by the game engine depending on how far away the player is standing from the object.

### 3.2 Objects for the Shed

As the artist, you must create objects in 3D Studio Max that follow specific naming conventions. You also need to create important sub-objects that tell the engine how to handle the object in the game.

The primary object must be given a name that is unique to the game. All sub-objects must be unique within the object.

In 3D Studio Max you'll see all the primary and secondary objects that are required to create a complete shed, including invisible objects that exist only to tell the engine where certain boundaries exist.

#### Primary object

The exterior walls of the shed are grouped together to create the primary object. The names of all the primary objects start with "01". Each "01" object name must be unique in the game.

When the engine sees a "01" name, the game engine knows the object is a primary one, and will render it. If the name of the primary object starts with a "02" (or something else), the object is a child of the primary object.

The exterior walls of this shed are named *01 extwalls*.

## Ground point

All buildings need a ground point, which is the contact point for the building.

The ground point for the shed is the box sitting inside on the floor. More accurately, the center of the ground point box is the actual ground point. By drawing a box that seems to be sitting on the floor, the actual ground point is positioned above the bottom of the shed walls. This positioning causes the bottom section of all buildings to be drawn under the ground. Burying the bottom of a building allows it to be placed on a sloped hill, without one edge of the building being suspended in mid-air.

The ground point sub-object of the shed is named *upg01*.

## Secondary objects

The names of secondary objects start with subsequent numbers. For example, all the interior walls of the shed are grouped into one object named *02 intwalls*. The object representing the room will contain the other objects that control game-play in the room.

The shed includes an example of another secondary object, a light. The light is named *LP02*. Naming a light with “LP” tells the Object Editor that the object is an object that provides light.

## Center point

Every sub-object that you create must be given a center point. For example, the secondary object of the interior walls needs a center point.

The typical name for a center point is *\_02 center*.

## Collision box

The collision boxes are added to all solid parts of your object so that other objects can't pass through them. For example, if you draw a wall, but you don't draw a collision box that covers the wall, then a player could run through it. Collision boxes also prevent people from falling through upper-story floors. You don't need a collision box for the terrain.

Collision boxes must be a convex shape; the engine will not handle concave shapes.

One of the collision boxes for the shed is *CB01a*. Other collision boxes on the same object are given subsequent names (e.g. *CB01b*). Collision boxes for other sub-objects are given appropriate names (e.g. *CB03a*).

### **3.3 Export the Object from 3D Studio Max**

When you finish creating your shed in 3D Studio Max object, you need to export it as a 3DS ASCII file so that the Object Editor (OED) can load it. The extension for a 3DS ASCII file is .ASE. In this example, the file is named `ShedH.ASE`, and it was provided with the mod tools release.

If you made any changes to the shed in 3D Studio Max, save and export the object to the same name so that the rest of the directions in the tutorial make sense. Keep in mind that the .ASE file name will be the same name used in the NovaLogic Object Editor (except for the extension). That way, when you make other changes in 3D Studio Max and export an updated .ASE file, the OED will import those changes automatically when you open the OED version of the file. This will save you a lot of time.

When you export the file, choose the working directory that was mentioned earlier. When you import the .ASE file into the OED, all your working files will be saved in the working directory and not affect the Black Hawk Down game files.

**Important:** Copy the `NLASCEXP.DLE` export plug-in file to the 3D Studio Max plug-in directory. This will allow you to export an .ASE file for the OED.

## 4 Edit the Shed in OED

Once you've exported an object from 3D Studio Max (e.g. `ShedH.ASE`), you can edit it in the NovaLogic Object Editor (OED). Using OED, you can add other game controls to the objects and sub-objects of the shed.

To start OED, run `df4oed` from your `ModTools` directory.

### 4.1 Import the 3D Studio Max File

Before importing a 3D Studio Max file into OED, make sure that the *1 unit = 1 m* box is selected in OED. If you don't tell OED to draw the object in meters, the object will be drawn in feet, and your object will be roughly one third the size of other game objects.

You also need to tell OED where to find your object texture files. From the **Tools** menu, open the **Option** window. Change the **Default Texture Folder** to your `ModTools` folder.

To import, open the **File** menu and click **Import 3DS ASCII**, and pick the `ShedH.ASE` file. OED will load the file and draw a wire-frame version of the shed.

After importing the `.ASE`, you should save the file. OED saves the object as a `.3DA` file. Both the `.ASE` file and `.3DA` file should be given the same name. This way, any changes you make to the 3D Studio Max file will be imported automatically the next time you open the object in OED; you won't need to re-import the `.ASE` file if you name them the same.

The OED performs some validation checking of the objects. When OED discovers an invalid object, the object is drawn in red.

### 4.2 Edit the Object

OED lets you change various settings of objects and add a variety of controls to them.

In this shed example, the primary purpose of OED is to export the `.3DI` file so it can be used in the NovaLogic Mission Editor (MED).

For more details about editing an object in OED, refer to the **OED Manual**.

### OED Basics

After importing the `.ASE` file you'll see a wire drawing of the shed. You can click the **Render** button to see the shed painted with textures.

Click the **3D** box to change the current method for viewing the shed.

Zoom into the shed by pressing the SHIFT key while holding down the right mouse button, and move the mouse from side to side.

You can slide the shed by pressing the SHIFT key while holding down the left mouse button and moving the mouse in any direction.



Rotate the shed by pressing the CTRL key while holding down the right mouse button, and moving the mouse.

## Adding a Fan

If this room looks a bit dull, you might want to draw a ceiling fan inside the shed. You could do this using 3D Studio Max, but you wouldn't be able to make it rotate in OED because the engine doesn't support nesting complicated objects like that. Instead, think about your objects more modularly.

Create a separate fan object 3D Studio Max, and then add rotating controls to the fan in OED. Later, in the NovaLogic Mission Editor (MED), you could add the fan to the shed, or other buildings.

**Tip:** OED buttons that look flat and have the word **None** in them can be clicked.



## 4.3 Export for Mission Editing

Once you've finished changing to the shed in OED, you need to save and then export the file. Exporting the file prepares it for being packed into your new mod file, and making it available to be added to a mission.

From the **Tools** menu, open the **Option** window. Change the **Default 3DI Export Folder** to your `ModTools` folder.

On the **File** menu, click **Export 3DI** and enter the file name, `Shed.3DI`.

The exported shed object is almost ready to be added into a Black Hawk Down mission.

## 5 Add Object to Item Definitions

Once you've exported an object from OED, you need to add it to the Black Hawk Down item definition file. The item definition file contains the object definitions for all the objects in the Black Hawk Down game.

In this example, we'll add the shed object to the `ITEMS.DEF` file provided in the `ModTools` directory. When we're done, the `ITEMS.DEF` file will be included in the packed file that is used to load your mod changes.

To add the shed to the list of game objects, open `ITEMS.DEF` in your favorite text editor, and insert the following lines into the end of the `ITEMS.DEF` file.

```
begin "Concrete Shed Test Building"
  id 102399
  type building
  graphic shed
  attrib: Landable
  render_function bldg
end
```

Once you've added the shed to the item definition file, you can insert the shed into a mission.

To add other types of objects, just find a similar object in the `ITEMS.DEF` file, copy those lines, and make changes to the attributes, especially the `id`. To pick a unique `id` for an object in the mod, make sure you pick a unique number that is greater than any existing number in the file.

**Important:** Make sure you pick a unique `id` for every object you add to `ITEMS.DEF`.

## 6 Insert Object into a Mission

Now that you've finalized the shed object, exported it from OED, and added the shed to the `ITEMS.DEF` file, you need to pack the files into your `.PFF` file so that you can insert the building into a Black Hawk Down mission.

From your `ModTools` directory, run the following commands to add the object files for the shed into the `.PFF` file for your mod.

```
pack MyMod.pff +Shed.3di
```

```
pack MyMod.pff +ITEMS.def
```

You'll next need to run the NovaLogic Mission Editor (MED) from the Black Hawk Down game directory. It is named `dfbhdmed`, but don't open it yet.

First, you need to create a shortcut, and pass in parameters so MED can find your new `.PFF` file. To do this, create a shortcut in your game directory, right-click on it, and pick **Properties** in the menu. In the **Target** field, add these two parameters after the last double-quote: `"/mod"` `"MyMod"`.

For example if you were to run MED from the command line, the command would be:

```
dfbhdmed /mod MyMod.
```

Now, run that shortcut (or command line), and it will automatically load the `.3DI` file of your shed. In MED, open one of your existing missions or create a new one. When you insert a building, MED will be able to find the new shed.

Here's a quick description of how to use MED to add a shed to a mission.

- Open one of your existing missions, or create a new one.
- Put MED into **Insert** mode (right-click in the drawing, and click **Insert Mode** in the mouse menu).
- Click on the map to pick the location for the shed. The **Insert Item** window will open.
- Select the **Type** of object you want to place. In this case, pick **Buildings**.
- Scroll down until you find the Shed object. Click it and MED will place the shed at the location you picked.

The MED manual on your Black Hawk Down game CD provides detailed information for creating missions and adding objects.

## 7 Pack the Mod

Your mission files and new object files need to be packed into one .PFF file that can be called by the Black Hawk Down game.

You can pack files with the `pack` application in your `ModTools` directory. In the following command, replace `MyMod` with the name of your mod. Replace `new_file` with the name of your mission.

```
pack MyMod.pff +new_file.bms
```

You've now created a .PFF file that contains all the files for your modification of Black Hawk Down. It's time to run it!

For more information about preparing files for your mod, read the **Pack tool Overview** document.

## 8 Run the Mod

Now that you've packed all the new or changed files into the .PFF file, you can run the mod, telling it the location of your new mod file. Typically, this you will build your .PFF file in the `ModTools` directory where you saved your mod files.

**Important:** Your final .PFF file must be stored in a directory named `mod` that is located in the Black Hawk Down game directory.

Run the following command from the game directory, either in a command window, or as a shortcut. Replace `MyMod` with the name of your mod.

```
dfbhd /mod MyMod
```

Pick your mission from the mission list, and check out your new object!