



## Building exporter tools

**Disclaimer**

**Building export tools are provided “as is”.**

**Monte Cristo DOES NOT make any support for those tools and shall not be held responsible for any inconvenience or mishap occurring as a result of the use of those tools.**

1 - Look how others buildings are made.....	2
2 - Create a building for City Life: from 3DSMax to City Life.....	2
Step 1: Export file from 3DSMax .....	2
Step 2: Content gathering.....	3
Step 3: Main.ini file creation.....	4
Main.ini file content example .....	4
Step 4: File creation .....	6

## 1 - Look how others buildings are made

In order to help you in beginning to make new buildings, you can look to the way other buildings are made. This will be more specifically useful for the **Main.ini** file so that you're sure about the structure of the file and of the .pkb file.

To do so:

- Rename the extension .pkb file it to .pak (for instance "NewBuilding.pkb" to "NewBuilding.pak")
- Then drag & drop the file on the file called **\_FileSystemPacker.exe** (this file is located in CityLife/DATA folder)
- This generates a .zip file (for instance "NewBuilding.zip"). This file can be unzipped.
- In the zip file you have all the content files (mesh, textures, main.ini, etc...) which allow you to see how the building was made.

It is recommended even for advanced user to study existing functional main.ini files before trying to create one.

## 2 - Create a building for City Life: from 3DSMax to City Life

*Warning: importing a new building is an easy and automated operation.*

**Creating a .pkb file, on the other hand, should be tried by advanced users only.**

The Building Importer program can only import buildings in a certain format.

This format is .pkb file.

This file must include all the information to install one or several new building(s) in the game.

If the .pkb file is incorrect or incomplete, the import operation will fail.

**NB: This document is for buildings which are NOT residential buildings.**

### Step 1: Export file from 3DSMax

Once you've installed the Monte Cristo plug-in and associated files, here is the way to export a building made with 3DSMax

- Open the Building file you want to export
- Select the building to export
- Right click on it and in the contextual menu select "Properties"
- In the window which appears, select the tab called "User Defined"
- In this window Copy/Paste the following line
- SHADER = MCBuilding
- Then Click on OK at the bottom of this window.

Then ...

- Click on "Utilities" (hammer icon on the top right of the screen)
- Click on the button called "More"
- In the list appearing, select "Export Template"
- Click on the button "OK"
- In the Window appearing, simply click on "Select". This closes the window.

Finally ...

- In the File Menu, click on "Export"
- There, in the File format, choose "MonteCristoGames export format"
- Name your file (for instance "Example.sg")
- Click on "Save"

All the accurate files are exported in the directory you chose.

## Step 2: Content gathering

For this section, we'll work with the following names:

- the working folder is called "NewBuilding" (Note: the name "NewBuilding" is given as an example, you can use a different name)
- the name of the building created here is "Example" (Note: the name "Example" is given as an example, you can use a different name)

Create an new folder called "NewBuilding".

Copy all the elements you created for your building to your "NewBuilding" folder.

- Elements exported from 3DSMax
  - o .sg files contains the name of the .mesh files and of the dummies
  - o .stdmat files hold material definitions, texture slot list (diffuse, opacity, ...) and texture names, contains the name of the .tga file used
  - o .tga files hold textures data (note that alpha channel is never used)
  - o .mesh contains geometry and material binding, and the names of the .stdmat and the .meshdata
  - o .meshdata contains raw geometry data (positions, texture coordinates...)

**NB: the export from 3DSMax generate files in subdirectories called Textures and Shaders. Here you have to place all those files in the same directory called here "NewBuilding"**

- Gameplay elements
  - o You will need to create a "prototype file", i.e. an .xml file holding all the game simulation information:
    - a prototype file is a file which gives to the City Life simulation engine all the information of what is the building and how it works. The file format is <filename>.xml
    - you can find the original prototype files for the game's buildings in \City Life\DATA\DESIGN\Buildings
    - To create your own prototype file:
      - Select an existing prototype of a building similar to the building you want to create.
        - o Example: to create a new supermarket, select the prototype file of the supermarket
        - o to create a new industry for the Elite, select the prototype file of an Elite industry
      - Copy the selected prototype file and paste it in the "NewBuilding" folder you created
      - Rename your prototype file to the name of your choice (this will not be the building's name in the game, it is only a codename)
      - Edit it (using Windows Notepad for instance):
        - o **1 – Define the Mesh to use**
        - o In the file, locate the line starting with <MeshState0> in the <ENTITY> section. For example  
<MeshState0>..\data\game\_data\Buildings\building\_name.sg</MeshState0>
        - o In that line, replace the .sg file name (here "building\_name.sg") by your file name (in this example, "Example.sg"):  
<MeshState0>..\data\game\_data\Buildings\_addons\Example.sg</MeshState0>
        - o **2 – Define the icon file (png file) to use**
        - o In the file, locate the line starting with <Icon> in the <MENU> section.  
For example  
<Icon> BOUT\_CONSTRU\_SERV\_APPRO0</Icon>
        - o In that line, replace the file name between the tags by the file name of your icon (the .png file). For instance  
<Icon>EXAMPLE</Icon>
      - Save your prototype file and close it.

- Interface elements
  - o You will need to create an “icon” file, i.e. an .png file that will be used as an icon representing the building in the game’s interface
    - This icon .png file must be a square of 56x56 pixels

### Step 3: Main.ini file creation

Now that you have all the files in your “NewBuilding” folder, you need to create a file that will allow the building importer to manipulate your data.

In other words, you need to create a file where you say “mesh is located there, textures are there, icon to use for this building in the game is there, etc...”

To do so, in your “NewBuilding” folder, you must create a file you will call **Main.ini** to declare all those information.

Once created, edit it with a text editor (Windows Notepad for instance).

In order that the program can find the accurate information, you have to follow a specific structure, and organise information in different sections called [S0],[S1],[S2], etc. (see below for details)

The Main.ini file is the file to be analysed by the Building importer program.

It describes where each file you have placed in the folder during Step 1 must be copied in the game folders hierarchy.

The Main.ini file is a collection of sections.

The important thing is that when you add a new information, you begin a new section. Section numbers should all be different. A given number is not associated to a given category of information.

Example: IconGraph could be section [S8] or [S22], or any other number, as long as this number is not duplicated anywhere.

Here is an example of Main.ini file complete with a short description of the sections:

### Main.ini file content example

- **BuildingGraph** (ex. [S0], [S1], [S2]) :  
The building importer will use this kind of section in the following manner: it will copy the files listed under “elements” in the “Path” folder. The path folder is considered starting in the CityLife/Data folder. Under “elements”, all the names of the .sg .mesh and .meshdata parts of the building

The line below for instance will copy the “Example.meshdata” file from the .pkb file to the CityLife/Data/Game\_Data/buildings\_addons directory

```
[S0]
Type          = BuildingGraph
Path          = Game_data/Buildings_addons
Elements      = Example.meshdata Example.mesh Example.sg Example_L1.meshdata Example_L1.mesh
Example_L2.meshdata Example_L2.mesh Example_L3.meshdata Example_L3.mesh
```

Section [S1] for instance copies the “Example\_diff.tga” texture file from the .pkb file to the CityLife/Data/Game\_Data/buildings\_addons/Textures directory

```
[S1]
Type          = BuildingGraph
Path          = Game_data/Buildings_addons/Textures
Elements      = Example_diff.tga Example_ill.tga Example_spec.tga
```

```
[S2]
Type          = BuildingGraph
Path          = Game_data/Buildings_addons/Shaders
Elements      = Example_L3Example_NS_.stdmat Example_L3Example_sol_NS_.stdmat
ExampleExample_NS_.stdmat ExampleExample_sol_NS_.stdmat Example_L1Example_NS_.stdmat
Example_L1Example_sol_NS_.stdmat Example_L2Example_NS_.stdmat Example_L2Example_sol_NS_.stdmat
```

- **BuildingXml** (ex. [S3]) :  
This section works the same way as the **BuildingGraph** section, i.e. it copies the .xml file used by the building to the accurate directory

[S3]  
Type = BuildingXml  
Path = Design/Buildings\_addons  
Elements = Example.xml

- **TreeRoot** (ex. [S4]) :  
Add to the **Tree\_Root.ini** file, the information which allows access to the new building in the Building interface. This is the place where you say “place a new icon in this place of the building tree”.  
This Section allow to declare in which category to store the new building (for instance, FRINGE\_INDUSTRY, etc.).  
The “AddPrototype” line adds the prototype file to the prototype list  
Place here the name of the file WITHOUT the extension, for instance : Example and NOT Example.xml

Caution : no graphical test verifies if there is enough place in the construction menu to add a new building.  
This is the part linked to the interface hierarchy (describe the icon’s category).

[S4]  
Type = TreeRoot  
Path = Design/Interface/Tree\_Root.ini  
Section = FRINGE\_INDUSTRY  
AddPrototype = Example

- **IconGraph** (ex. [S5]):  
This section works the same way as the **BuildingGraph** section, i.e. it copies the .xml file used by the building to the accurate directory. Here it copies the icon of the building in the Building interface.

[S5]  
Type = IconGraph  
Path = Interface/overlay\_graph/icon\_addons  
Elements = Example.png

- **Overlay** (ex. [S6]) :  
Allow to add a section in **overlay\_graphics.ini** to declare the icons of new buildings which are in the list. This is the graphical declaration for the icon, describing its size and where the file for the icon can be found.

[S6]  
Type = Overlay  
Path = Overlay/overlay\_graphics.ini  
Section = EXAMPLE  
HEIGHT = 56  
WIDTH = 56  
X = 0  
Y = 0  
NB\_OBJECTS = 1  
FILENAME = DATA\..\DATA\INTERFACE\OVERLAY\_GRAPH\ICON\_ADDONS\TEST\_BAT.PNG\EXAMPLE.PNG

- **LaunchBat** (ex. [S12]) :  
Allow to execute a .bat application, for data cleaning, allowing the game application to regenerate data including the new building, etc.  
**Those two lines are to be inserted in the Main.ini file**

[S7]  
Type = LaunchBat  
Path = \_Delprotodat.bat

This will for instance delete the protodat file, a file that holds the compiled information of all prototype files. Deleting this file is necessary to take the new prototype into account. The protodat file is generated at game launch when the game does not find it. Regenerating the protodat

[S8]  
Type = LaunchBat  
Path = \_DelVirtualTable.bat

This will delete the virtual table, a data structure allowing the game to localize your game data on your harddrive. This data structure will be regenerated if it is not found at game launch.

#### Step 4: File creation

When the Main.ini file is created and copied to the folder generated during Step 1, drag and drop this folder on the file called *\_FileSystemPacker.exe*.

This file is located in CityLife/DATA folder (it is installed during game installation)

This operation will create a new file called "Example.pak"

Rename the file extension from .pak to .pkb to have a file called "Example.pkb"

You can now use this file with the "City Life building importer" application to import your new building in the game. You can also distribute this file to your friends so that they import your building in their City Life.