



Using the ResourceCompiler

2004-09-08

ResourceCompiler Files

The ResourceCompiler consists of the following files in the MasterCD directory:

Filename	Purpose
RC.exe	This is the executable which can be run from the command line.
ResourceCompilerPC.dll	This Plugin process and converts source files (CFG, CAF) to a destination file.
ResourceCompilerImage.dll	This Plugin process and converts source images files (TGA, DIB, JPG, PNG, DDS) to a destination file. The typical destination file format is the compressed DDS. The plugin is supposed to be used for several platforms.
RC.ini	This INI file stores some global parameters for the compile process

Command line usage of the ResourceCompiler:

The first parameter in the command line specifies the source. You can specify here a directory (e.g. C:\input or input/back), a directory with a pattern (e.g. C:\input*.tga) or a file name (e.g. input/test.tga or test.tga). The following parameters have to start with /, the key name (case sensitive) followed by the colon or the equal sign and the value you want to specify

Examples

- RC D:\input\ranger_statue.tga /p:PC /userdialog=0
- RC D:\input*.tga /p:PC /targetroot=d:\output /wait
- RC D:\input*.tga /p=PC /targetroot=d:\output /logfiles

Parameters

The compile process is controlled by parameters, which are set in INI files or via the command line.

1. Initial parameters come from the file **RC.ini**.
2. **File specific parameters** are stored in a file in the same directory as the source, with the same name but with extension **_RC**. The internal structure of the file is similar to INI files.
3. Further parameters can be specified in the **command line**.

Parameters from 1 are overwritten by 2 which are overwritten by 3. Command line parameters override parameter settings made in INI or **_RC** files.

General usage of the ResourceCompiler:

The following parameters are used:

Parameter	Usage
refresh	When present, causes files to be converted even if an up-to-date converted file already exists.
quiet	Do not write log to into console window.
p	Specifies the platform, can be one of: PC, XBOX, PS2, GC. Commandline example:

	/p:PC
recursive	1=process the source subdirectory recursively. 0=no recursion.
targetroot	specifies a destination directory where all compiled files are written to. Reproduces identical directory structure, creating directories as necessary. When this parameter is not specified, the targetroot is the same as the source directory, beware! Examples of usage: rc c:\mastercd /targetroot:c:\mastercd2 (creates a second mastercd folder with all converted files). rc c:\mastercd\textures\blah.jpg /targetroot:c:\mastercd2\textures (converts 1 file, and makes sure it ends up in the correct destination).
wait	Makes the ResourceCompiler wait for a keypress before exiting (useful when run in a situation where the output window disappears on program exit).
ext_E	Specifies special action to be taken for extension E. Valid actions are ignore to not process this kind of file at all, and copy to copy this file to the destination directory if it is different from the source (as specified by targetroot). For example: rc c:\mastercd /ext_log:ignore /ext_dll:copy makes sure that log files are not processed, and any dlls are copied to the target.
statistics	When present, create statistic files after compilation (useful for optimizations), like rc_stats_filedependencies.log rc_stats_materialdependencies.log
logfiles	When present, create log files during compilation (useful after a crash or for fixing all errors/ warnings), like rc_log.log (all messages) rc_log_warnings.log (only warnings) rc_log_errors.log (only errors)

Usage of ResourceCompilerPC

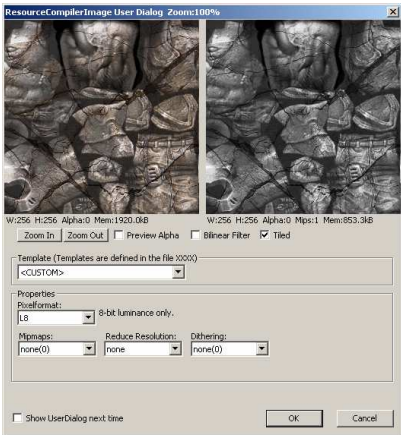
The following parameters are used:

Parameter	Usage / Values
GeomName	****
AnimatedCGF	Can be 1 or 0, if 0 animated cgf s will be skiped and not processed.
Stripify	When set to 1, the compiled cgf will be stripified.
LoadAdditinalInfo	When set to 1, the compiled cgf will contain node names and helpers.
KeepInLocalSpace	When set to 1, the compiled cgf will contain vertices in local space.

Usage of ResourceComplerImage:

The following parameters are used:

Parameter	Usage / Values
pixelformat	R8G8B8,A8R8G8B8,X8R8G8B8,R5G6B5, ... (open the user dialog to see all supported pixel formats)
mipmaps	0=don't create mipmaps. 1=create the full mipmap chain (+1/3 more memory).

mipmirror	0=texture is assumed as tiled for mipmap generation. 1= texture is assumed as mirrored for mipmap generation.
userdialog	0=don't show user dialog. 1= show user dialog. 
reduce	0=don't reduce resolution. 1=reduce resolution in x and y by 2. 2= reduce resolution in x and y by 4.
dither	This value is only used if the source color has to be reduced (e.g. with indexed palette). 0=no dithering. 1=simple dithering.

Future plans for ResourceCompiler

- Dependencies checks and build based on dependencies
- Presets
- Convert normalmaps
- Apply grayscale bump map to normal map
- Specify mipmap filter and quality
- Improve mipmap generation quality (optimized for 2x2 or 3x3)