

Output-Commander

(C) Lubo [Jannik Luboeinski] 2006 - jannik.lubo@web.de; www.lubo-online.de
version 2.0, 7th October 2006 (Russian translation by Vam)

General

Output-Commander is a helping tool for the mod development kits of the games Gothic and Gothic II. It reads the output units from ZenGin output files (.csl and .bin) and shows them in a list. These outputs can be recorded with the internal recording function or with an external recording program. Output-Commander supports the GOF file format developed by Caphalor and Lubo and it is able to export into different plain text formats.

Features

The name of the NPCs, the Voice-ID and the name of the relating sound file are shown. By turning off the showing of SVM outputs and by the marking of already recorded files it is much easier for modders to find writing mistakes and to have a sorted view of the speech recordings.

The sound files can be played, recorded or deleted by one mouse-click. Using the sort input you can get fast to a searched output. To search more closely, the universal search can be used, which can find optionally the ID, the output or the VoiceID. It is also possible to type an own structure for IDs. For example: *GIL_NID_NAM_***_VID_NUM* is the ID for normal dialogs. *GIL* specifies the guild, *NID* the NPC-ID, *NAM* the name. Next are following three stars, which are standing for a block of own output names. The elements behind the stars are read from the right side. *VID* specifies the VoiceID. *NUM* can be used as a wildcard for informationless blocks. Example: *Mnw_004_Ulbert_FOODGIVE_MISS_1_01*. Behind *FOODGIVE_MISS* could stand *_JA* e.g., because it is in the stars area.

Options

Speech samples path: the folder where the recorded files are stored

ID structure: see "Features"

Audiorecorder: a user-defined program to record speech samples

Pass an empty WAV file to the program: passes a new created WAV file with the output name to the recording program

Auto-Refresh for built-in recording: refreshes the list automatically after a file has been recorded with the internal recording function

Plugins

Output-Commander has a built-in plugin feature. It supports two types of plugins, format plugins and DLL plugins. Both must be copied to the subfolder \plugins for usage.

DLL plugins [.dll]:

DLL plugins are hardcoded plugins, which are used to extend Output-Commander. They must exist in form of DLLs with standard-call functions. The pre-defined functions, which are called from the DLL, can be looked up in the "Beispiel-Plugin". DLL plugins give the possibility to use and modify all global variables of Output-Commander, to receive window messages or to adopt any other tasks.

Format plugins [.fmt]:

This type of plugins are simple plain text files, which can be edited with a normal text editor. Their structure is built like the following:

name=format name
extension=the format's file extension
format=the structure for one Output Unit
s1=first used string
s2=second used string
s3=third used string
s4=fourth used string
s5=fifth used string

In the “format=” string can be used the following symbols:

%s = string
%% = percent character
%[x] = Wildcard for import, exports puts out [x]
\a[xxx] = [xxx] are three digits which specify an ASCII character
\0 = nullbyte
\t = tabulator
\n = line break
\\ = backslash

The “s[x]“ values are used for the %s symbol. The first %s means “s1=“. For “s[x]=“ can be used the following constants:

npc (ID des NPCs)
outputid (ID des Output-Units)
file (Dateiname der WAV-Datei)
voice (Voice-Nummer)
outputtext (Untertitelausgabe)

File formats (extendable through plugins)

- Import -

ZEN Binary Safe (*.bin)

ZEN ASCII for OUs (*.csl)

Gothic Output-Unit File (*.gof)

- Export -

Gothic Output-Unit File (*.gof)

Text file (2 types, *.txt)

Have fun with the program,
Lubo