This file defines what models and submodels to prefetch for each map. Prefetching helps to reduce stuttering, related to loading of those models and submodels on the fly. You can determine what submodels to put on the list by enabling developer mode, playing the game and looking for “Loading sequence: models/%model_name%.dol” (to enable developer mode put these lines in “config.cfg”: “sv_cheats 1”, “developer 1”).

This file can only recommend to load certain models/submodels to the game. The game can ignore some items from precaching if amount of free RAM is insufficient.

Wrong “extraprecache.epc” file can cause problems: if you use some custom models and leave original *.EPC file untouched then the game can try to load non-existent model file, and that always results in crash. To stop crashing on custom models you can clear *.EPC file with 0x00 bytes in hex editor or build correct precache file.

1 EPC file structure

Structure of “extraprecache.epc” file is shown on fig. 1.

Offset (bytes)	Section
0	Items count
4	Items list
4 + (precache list size)	Map count (for precache list)
4 + (precache list size) + 4	Precache list

Fig. 1

Item count defines how many items are present in items list.

Items list consists of map and model name text strings.

Map count defines how many maps are present in precache list.

Precache list defines links between maps and models.

1.1 Items list

Each item in the items list consists of char count field and null terminated string (Fig. 2).

Offset (bytes)	Section
0	Char count
4	Null terminated string
4 + (char count)	

Fig. 2 – structure of an item

Char count defines how many bytes are occupied by the text string. Occupied space should be equal to a multiple of 4.

The text string contains either map name without ”.bs2” (“c0a0”) or model name (“models/scientist.dol”).

1.2 Precache list

Precache list links maps and models by specifying connections between indexes of their text strings and it also allows to select submodels of any given model.

Structure of precache list entry is shown on Fig. 3.

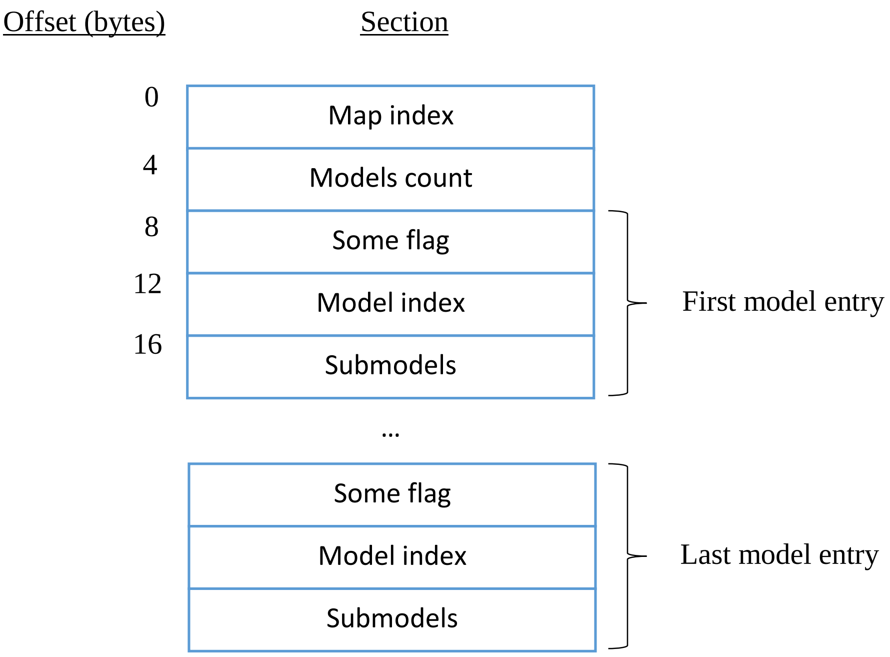

Fig. 3 – precache list entry

Map index corresponds to a number of the map item in the items list.

Models count defines how many model entries are linked to the map (and thus how many model entries are present in this precache list item).

Each model entry consists of three fields: Flag, Model index, Submodels.

I wasn’t able to determine a purpose of the Flag field. The only thing that I know for sure is that this flag is set for all models that are linked to Head-to-Head multiplayer maps.

Model index corresponds to a number of the model item in the items list.

The last field determines what submodels to prefetch. If it is nulled out then only model from model name text string would be precached. Every set bit inside this field can tell what submodels to load. You can see how it works on this example:

Model name string – “models/scientist.dol”;

Submodels = 0x0416 = 0000 0100 0001 0110 b;

Bit number	15	14	13	12		11	10	9	8		7	6	5	4		3	2	1	0
Bit value	0	0	0	0		0	1	0	0		0	0	0	1		0	1	1	0
Submodel #	-	-	-	-		-	10	-	-		-	-	-	04		-	02	01	-

For this case, those files would be precached: scientist.dol, scientist01.dol, scientist02.dol, scientist04.dol and scientist10.dol.