How to create a Gblorb file

In the following procedures, ***filename*** is the name of your game file. Just replace this with the actual name of your file.

Before you start

Download **https://www.ifarchive.org/if-archive/programming/blorb/iblorbb.zip**, unzip it and store the files somewhere, such as your Inform 6 **bin** folder. We will be using some of the exe files. These are MS-DOS executables, but they should work fine with Windows. I do not know of equivalents for Linux or Mac.

Step 1: Create resource file

1. Use your favourite text editor to create a resource file named ***filename*.res**. This should be formatted as follows:

```
CODE filename.z5|ulx
!Mandatory. This is the Z-code or Glulx file.
COPYRIGHT text
!Optional. This is the copyright statement for the resource file.
AUTHOR text
!Optional. Author's name.
NOTE text
!Optional. Any other information you want to include about the
resources.
RELEASE number
!Optional. Release number.
PALETTE palette_filename
!Optional. The palette to be used in the blorb palette format.
RESOLUTION resolution_filename
!Optional. The resolution to be used in the blorb resolution table
format.
HIGHCOLOR
!Optional. If included, this instructs to use a 16-bit palette.
TRUECOLOR
!Optional. If included, this instructs to use a 32-bit palette.
FRONTISPIECE cover_filename
!Optional. This directive acts identically to PICTURE, but also
generates a blorb frontispiece (Fpsc) chunk referring to that picture.
The constant FRONTISPIECE_ib is defined as a synonym for the name.
HEADER header_filename
!Optional. This includes the listed file as an IFhd (header) chunk.
METADATA metadata_filename
!Optional. This includes the listed file as an IFmd (metadata) chunk.
PICTURE image_ID image_filename
!Optional. Include one entry for each image.
SOUND sound_ID sound_filename
!Optional. Include one entry for each sound.
```

2. Compile the resource file as follows:

```
bres.exe filename.res
```

This will create two files:

- *filename*.**bli** is the file with all the constants that is included in your Inform 6 source code. (Remember 'i' for 'Inform'.)

- *filename*.**blc** is the control file that is used when you compile the gblorb file. (Remember 'c' for 'compile'.)

Step 2: Create Glulx file

1. Use your favourite source code editor to create your Inform 6 source code as usual.
2. Add the following `Constant`s before including **parser.h**, omitting any that are not relevant to your game:

```
Constant GG_GRAPHWIN_ROCK 210; !Graphics handle
Constant GG_BACKGROUND_CHANNEL_ROCK 410; !Background sound handle
Constant GG_FOREGROUND_CHANNEL_ROCK 411; !Foreground sound handle
Constant BACKGROUND = 0;
Constant FOREGROUND = 1;
```

3. Add the following `Global`s before including **parser.h**, omitting any that are not relevant to your game:

```
Global gg_graphwin = 0; !Graphics window number (0 = none)
Global current_pic = 0; !Graphics file id (0 = none)
Global gg_background_channel = 0; !Background sound channel (0 = none)
Global gg_background_repeat = 1; !Background sound repeats (0 = none,
-1 = loop)
Global gg_foreground_channel = 0; !Foreground sound channel (0 = none)
Global gg_foreground_repeat = 1; !Foreground sound repeats (0 = none,
-1 = loop)
Global background_sound = 0; !Background sound file id (0 = none)
Global foreground_sound = 0; !Foreground sound file id (0 = none)
```

4. Include the following files after including **verblib.h**:

```
Include "infglk.h";
Include "filename.bli";
```

5. Add a few more constants after the `Include`s as follows:

```
Constant ACTIVE 0;
Constant GRAPHICS 1;
Constant SOUND 2;
Array media_state -> 3;
```

6. Add the following in your `Initialise` routine, omitting anything not relevant to your game and making adjustments for things like image size:

```
if (media_state->ACTIVE == false)
{
  @protect media_state 3;
  media_state->ACTIVE = true;
  media_state->GRAPHICS = true;
  media_state->SOUND = true;
}
if (glk_gestalt(gestalt_Graphics, 0) && media_state->GRAPHICS == true)
{
  if (gg_graphwin == 0)
  {
    gg_graphwin = glk_window_open(gg_mainwin, winmethod_Above |
winmethod_Fixed, 240, wintype_Graphics, GG_GRAPHWIN_ROCK);
  }
}
if (glk_gestalt(gestalt_Sound, 0) && media_state->SOUND == true)
{
  if (gg_background_channel == 0)
    gg_background_channel =
glk_schannel_create(GG_BACKGROUND_CHANNEL_ROCK);
```

```
  if (gg_foreground_channel == 0)
    gg_foreground_channel =
glk_schannel_create(GG_FOREGROUND_CHANNEL_ROCK);
}
```

7. Add the following entry-point routines:

```
[ IdentifyGlkObject phase type ref rock id;
  if (phase == 0)
  {
    gg_graphwin = 0; !Graphics off
    gg_background_channel = 0; !Background sound off
    gg_foreground_channel = 0; !Foreground sound off
    return;
  }
  if (phase == 1 && type == 0 && rock == GG_GRAPHWIN_ROCK)
  {
    gg_graphwin = ref;
    return;
  }
  if (phase == 2)
  {
    DrawGraphics();
    id = glk_schannel_iterate(0, gg_arguments);
    while (id)
    {
      switch (gg_arguments-->0)
      {
        GG_BACKGROUND_CHANNEL_ROCK:
          gg_background_channel = id;
          PlaySound(BACKGROUND, gg_background_repeat);
        GG_FOREGROUND_CHANNEL_ROCK:
          gg_foreground_channel = id;
          PlaySound(FOREGROUND, gg_foreground_repeat);
      }
      id = glk_schannel_iterate(id, gg_arguments);
    }
    PlaySound();
  }
];
[ HandleGlkEvent ev context;
  context = 0; ! suppress ignored warning
  switch (ev-->0)
  {
    evtype_Redraw, evtype_Arrange:
      DrawGraphics();
    evtype_SoundNotify:
      if (ev-->3 == 1)
      {
        background_sound = 0;
      }
      if (ev-->3 == 2)
      {
        foreground_sound = 0;
      }
  }
];
```

8. If your game includes graphics, add the following routine:

```
[ DrawGraphics pic_width win_width x;
  if (glk_gestalt(gestalt_Graphics, 0) && media_state->GRAPHICS ==
true)
  {
    if (gg_graphwin ~= 0)
```

```
    {
      glk_window_clear(gg_graphwin);
      glk_image_get_info(current_pic, gg_arguments,
gg_arguments+WORDSIZE);
      pic_width = gg_arguments-->0;
      glk_window_get_size(gg_graphwin, gg_arguments,
gg_arguments+WORDSIZE);
      win_width = gg_arguments-->0;
      x = (win_width - pic_width) / 2;
      if (x < 0)
        x = 0;
      glk_image_draw(gg_graphwin, current_pic, x, 0);
    }
  }
];
```

9. If your game includes sound, add the following routine:

```
[ PlaySound channel repeat;
  if (glk_gestalt(gestalt_Sound, 0) && media_state->SOUND == true)
  {
    if (channel == BACKGROUND && gg_background_channel ~= 0)
    {
      gg_background_repeat = repeat;
      if (background_sound == 0)
        glk_schannel_stop(gg_background_channel);
      else
        glk_schannel_play_ext(gg_background_channel, background_sound,
repeat, 1);
     }
    if (channel == FOREGROUND && gg_foreground_channel ~= 0)
    {
      gg_foreground_repeat = repeat;
      if (foreground_sound == 0)
        glk_schannel_stop(gg_foreground_channel);
      else
        glk_schannel_play_ext(gg_foreground_channel, foreground_sound,
gg_foreground_repeat, 2);
     }
  }
];
```

10. If your game includes graphics and you want to be able to turn them on and off, extend the grammar as follows:

```
Verb meta 'graphics'
  * -> Graphics
  * 'on' -> GraphicsOn
  * 'off' -> GraphicsOff;

[ GraphicsSub;
  if (~~glk_gestalt(gestalt_Graphics, 0))
    "Your interpreter doesn't support graphics.";
  print "Your interpreter supports graphics and graphics are currently
";
  if (media_state->GRAPHICS == true)
    "on.";
  else
    "off.";
];

[ GraphicsOnSub;
  if (~~glk_gestalt(gestalt_Graphics, 0))
    "Your interpreter doesn't support graphics.";
  if (media_state->GRAPHICS == true)
    "Graphics are already on.";
```

```
  media_state->GRAPHICS = true;
  gg_graphwin = glk_window_open(gg_mainwin, winmethod_Above +
winmethod_Fixed, 240, wintype_Graphics, GG_GRAPHWIN_ROCK);
  DrawGraphics();
  "Graphics are on.";
];

[ GraphicsOffSub;
  if (~~glk_gestalt(gestalt_Graphics, 0))
    "Your interpreter doesn't support graphics.";
  if (media_state->GRAPHICS == false)
    "Graphics are already off.";
  media_state->GRAPHICS = false;
  glk_window_close(gg_graphwin, gg_arguments);
  gg_graphwin = 0;
  "Graphics are off.";
];
```

11. If your game includes sounds and you want to be able to turn them on and off, extend the grammar as follows:

```
Verb meta 'sound'
  * -> Sound
  * 'on' -> SoundOn
  * 'off' -> SoundOff;

[ SoundSub;
  if (~~glk_gestalt(gestalt_Sound, 0))
    "Your interpreter doesn't support sound.";
  print "Your interpreter supports sound and sound is currently ";
  if (media_state->SOUND == true)
    "on.";
  else
    "off.";
];

[ SoundOnSub;
  if (~~glk_gestalt(gestalt_Sound, 0))
    "Your interpreter doesn't support sound.";
  if (media_state->SOUND == true)
    "Sound is already on.";
  media_state->SOUND = true;
  glk_schannel_unpause(gg_background_channel);
  "Sound is on.";
];

[ SoundOffSub;
  if (~~glk_gestalt(gestalt_Sound, 0))
    "Your interpreter doesn't support sound.";
  if (media_state->SOUND == false)
    "Sound is already off.";
  media_state->SOUND = false;
  glk_schannel_pause(gg_background_channel);
  foreground_sound = 0;
  glk_schannel_stop(gg_foreground_channel);
  "Sound is off.";
];
```

12. Every time you need to replace the image, set `current_pic` to the *image_ID*, then call `DrawGraphics()`. For example, in the `description` for your room, use something like this:

```
description
[;
  current_pic = room01_pic;
  DrawGraphics();
```

```
    "You're in the test lab.";
],
```

13. Every time you need to play a background sound (such as theme music), use something like this:

```
background_sound = death_sound;
PlaySound(BACKGROUND, 1);
```

14. Every time you need to play a foreground sound (such as a squeaking door), use something like this:

```
foreground_sound = whistle_sound;
PlaySound(FOREGROUND, 1);
```

15. Compile your Inform 6 file as follows:

```
inform.exe +include_path=.\,your_lib_path -G filename.inf
```

This will create your ***filename*.ulx** file.

Step 3: Create Gblorb file

1. Bundle all the resources from steps 1 and 2 as follows:

```
blc.exe filename.blc filename.gblorb
```

This will create your ***filename*.gblorb** file.