Freeware Sound Gauge - dsd_xml_sound3.gau

This gauge represents a significant improvement (I think) over the initial sound gauge I released a year ago. The logic in the gauge has been improved, so you shouldn't hear extraneous sounds on start-up. The dependence on GaugeSound.dll has been eliminated by using DirectX calls. The ability to read XML Lvars has eliminated the need for FSUIPC and the ADF2 variable.

dsd_xml_sound3.gau uses a separate Lvar for each sound it controls. This approach costs a little more in clock cycles, because the gauge must monitor each Lvar, but the hit wasn't noticeable on my 1.4 GHz machine.

The gauge has a single RGB (0,0,0) bitmap, so you can place it anywhere on your panel. Keep in mind that like most other gauges, if you put it on a panel window that isn't open at panel load time, it won't work until the panel window is visible.

The gauge checks for the existence of a sound file before attempting to play it. If the gauge doesn't find the file, it will take no further action, but it won't crash the sim!

A Word About Config Files
This gauge can (optionally) use a configuration file to customize the sounds being played, along with the Lvar(s) controlling them.

You can specify the configuration file by way of a parameter in panel.cfg, thereby allowing for as many different configurations as desired.

The gauge will not make any modifications to the string you supply as a parameter, so you will need to ensure that you supply correct path information.

The 'GetPrivateProfileString' command that I used to read values in from the configuration files has one peculiar quirk - if you specify a filename without path information, it will look for that file in the main Windows directory, not the directory where the application (FS9) is running. The minute you specify a path in front of your filename, the 'GetPrivateProfileString' command will look in your FS9 folder for the folder specified in the parameter. Some examples:

This is how NOT to do it – in this example the gauge will look for config_file.ini in the Windows directory:
gauge03=dsd_xml_sound3!dsd_xml_sound3, 585,278,20,20, config_file.ini

Placing <./> in front of the filename instructs the gauge to look in the main FS9 folder for config_file.ini:
gauge03=dsd_xml_sound3!dsd_xml_sound3, 585,278,20,20, ./config_file.ini

This is the equivalent of the default (no parameter specified) scenario:
gauge03=dsd_xml_sound3!dsd_xml_sound3, 585,278,20,20, ./Gauges/dsd_xml_sound.ini

The gauge will look in the FS9 gauges folder for the file named dsd_xml_sound.ini

While it is possible to specify an absolute path including a drive letter, it is not a good idea because the user's installation might not be in the same place as yours is. You would be better off with this:
gauge03=dsd_xml_sound3!dsd_xml_sound3, 585,278,20,20, .\Sound\Piper_sounds.ini
This example specifies that the file Piper_sounds.ini is located in the FS9\Sound folder.

Note that the </> and <\> characters both indicate folders in these parameter strings.

In summary, specify <./> at the start of your parameter to root the string in the main FS9 folder, and specify sub-folders from there.

This gauge uses multiple XML Lvars to accept commands to play sounds. The Lvars used, and the sounds to be played, can be specified in a text-based initialization file. The path to, and name of, the initialization file can be specified as a parameter in the gauge's entry in panel.cfg - a different configuration is therefore possible for every panel the gauge is used in.

This gauge uses a separate Lvar for each sound to be controlled. Meaningful values for the Lvar are:

0 - stops the sound if it is currently playing.
1 - plays the sound once. The gauge will set the Lvar back to 0 when the sound finishes or is stopped.
2 - 'loops' the sound. The gauge will continue to play the sound until your XML gauge sets the related Lvar to 0.
3 – sets the sound's volume level.
4 – retrieves the sound's current volume level.

Default Values
Initialization File: Located in the Gauges folder and named dsd_xml_sound.ini. Note that the use of an initialization file is not required. If you don't use one, the gauge will use default values for Lvars and sound file locations and names. These defaults are:

Lvars: dsd_xml_sound_id_xx, where xx is a two-digit number between 00 and 99

File names: dsd_xml_xx.wav, where xx represents the same two-digit number as above, from 00 to 99

File locations: \Sound\dsd (Folder dsd contained in the main FS Sound folder)

The default values tend to imply that there is a numbering requirement in the Lvar names and sound file names, but this is not the case. You are free to choose any Lvar names and sound file names you wish. The relationship between the Lvar and the sound file is established by the values to the left of the equal sign in the .ini file. Two sample .ini file extracts should help clarify this:

```
[LVars]
Lvar00=dsd_xml_sound_id_00
Lvar01=dsd_xml_sound_id_01
Lvar02=dsd_xml_sound_id_02

[Sounds]
Sound00=./Sound/dsd/dsd_xml_00.wav
Sound01=./Sound/dsd/dsd_xml_01.wav
Sound02=./Sound/dsd/dsd_xml_02.wav
```

Using default values, the above Lvars and sounds establish that dsd_xml_sound_id_00 will control dsd_xml_00.wav; dsd_xml_sound_id_01 will control dsd_xml_01.wav; and dsd_xml_sound_id_02 will control dsd_xml_02.wav. Referring to the values on the left of the equal signs, Lvar00 controls Sound00; Lvar01 controls Sound01, and Lvar02 controls Sound02.

If you customize the .ini file, it might look like this:

```
[LVars]
Lvar00=Ignition_Switch
Lvar01=landing_gear_up
Lvar02=landing_gear_down

[Sounds]
Sound00=./Sound/click.wav
Sound01=./Sound/new_l1011/gear_up.wav
Sound02=./Sound/new_l1011/gear_down.wav
```

In this example, the Lvar L:Ignition_Switch will control the sound file click.wav, located in the main sound folder.

L:landing_gear_up will control the sound file gear_up.wav, located in a folder called new_l1011, which is contained in the main sound folder.

L:landing_gear_down will control the sound file gear_down.wav, located in the same new_l1011 folder.

[Config]
This section of the .ini file can be used to specify the following configuration options: MaxSounds, LvarStop, VolumeVar and ResetVolumeVar.

MaxSounds. This parameter can reduce the number of sounds the gauge will control. The gauge has a hard-coded maximum of 100 sounds, but you can reduce that to a lower limit if you wish; doing so will save a few CPU cycles. The gauge scans each Lvar on each gauge update cycle. There certainly isn't anything to be gained by having the gauge scan a bunch of Lvars that are not going to be used to control sounds. When I tested the gauge, there really didn't seem to be a noticeable hit on frame rates (on a 1.4 GHz TBird), but the limiter is there if you wish to use it.

LvarStop. The Lvar specified here will stop all currently playing sounds if set to a value greater than zero. The gauge will reset the value to zero if the command is executed.

VolumeVar. The volume setting variable may be specified in the .ini file.
Put the name you wish to use on the VolumeVar= line.

ResetVolumeVar. The volume setting variable may also (optionally) be reset to -1 when it is used to change a sound's volume level. If you do not wish this reset to take place, set the ResetVolumeVar= line to 0. The default for this option is 'on'. That is, unless you specify otherwise, the volume setting variable will be reset to -1 every time it is used to change the volume level of a sound. The gauge will now check this variable when a play command is issued, and will use a non-negative value to set sound volume accordingly.

For example:
95 (>L:dsd_xml_sound_volume, number)
1 (>L:dsd_xml_sound_id_01, number)
will cause the volume level to be set to 95 immediately before the sound is played.

The gauge will now mute playing sounds when the sim is paused - the same procedure is also followed when FS's sound is turned off.

Panel.cfg Syntax Examples:
Example 1:
gauge03=dsd_xml_sound3!dsd_xml_sound3, 2,2,5, 5,

That is the default setting. The gauge will look in the gauges folder for dsd_xml_sound.ini, and will read that file if found. If not, default values will be used for Lvar and sound file names.

Example 2:
gauge03=dsd_xml_sound3!dsd_xml_sound3, 2,2,5, 5, .\Gauges\jet_sounds.cfg

The gauge will load the values from the file jet_sounds.cfg if found in the gauges folder. If that file is not found, defaults will load.

Example 3:
gauge03=dsd_xml_sound3!dsd_xml_sound3, 2,2,5, 5, .\Aircraft\b737_400\boeing_sounds.dat

The gauge will look for a file called boeing_sounds.dat in the main folder for the default 737. Again, if the file isn't found, default values will be used.

Setting the Volume
The gauge will accept a sound volume specification in the .ini file. To specify a volume level, add a number between 0 and 100 to the end of the sound file listing in the .ini file, like this:

Sound00=./Sound/dsd/dsd_xml_00.wav, 97.5
Sound01=./Sound/dsd/dsd_xml_01.wav, 82

A value of 100 represents full volume, while a value of 0 represents a 100-decibel attenuation. In practical terms, any value of less than about 70 will render the sound inaudible. Useful values will likely be in the 80-100 range. The default is 100, or no attenuation. DirectSound does not support amplification.

To set the volume of a sound, set L:dsd_xml_sound_volume to the desired volume, as described above. Then write a value of 3 to the sound's control variable. Setting volume level in this manner can be done on the fly, while the sound is playing.

For example:
95 (>L:dsd_xml_sound_volume, number)
3 (>L:dsd_xml_sound_id_01, number)
will cause the volume level to be set to 95.

Finding Out Current Volume
To find out the current volume level, write a value of 4 to the sound's control variable. The current volume level will be assigned to L:dsd_xml_sound_volume. This function can also be performed while a sound is playing.

For example:
4 (>L:dsd_xml_sound_id_01, number)
Left/Right Channel Balance
I have made use of the DirectSound SetPan function to allow you to manipulate left/right sound balance. This function can be performed only when the initial command to play the sound is given. Here is how to do it:

The range from maximum left to maximum right is specified as a number between -10000 (full left) and 10000 (full right).
- Multiply the absolute value of the desired setting by 100.
- Add the result to the desired control variable setting (1 for play once, 2 for loop).
- Make the number negative if a left pan is required.
- Write this value to the control variable.

A couple of examples...
First, a slight left pan on a looping sound (-1000):
- Absolute value of the pan setting is 1000
- Multiply that value by 100 to get 100000
- Add this value to the desired control variable setting to get 100002
- Make the number negative because this is a left pan: -100002
- Write this value to the control variable:

-100002 (>L:dsd_xml_sound_id_01, number)

Second example, a more extreme pan to the right for a play-once sound: - 3000 is probably enough to eliminate all sound from the left channel.
- Absolute value is 3000
- Multiply by 100 to get 300000
- Add this value to the desired control value of 1 to get 300001
- This is a right pan, so the number stays positive
- Write the result (300001) to the control variable:

300001 (>L:dsd_xml_sound_id_01, number)

FS Sound Status
The gauge monitors the status of FS's own sound. You will find this value in L:dsd_sound_indicator. Further, when the FS sound is disabled, the gauge will internally set all sound volume levels to 0, so that sounds played by the gauge will behave in the same way as FS's own sounds. Note that this is a volume setting procedure - the sounds will still be playing, just very quietly. You may want to consider having your own gauge monitor L:dsd_sound_indicator and shutting off looping sounds, if practical, because playing them will still be consuming CPU cycles.

Nothing to do with sound, but the gauge will now use the variable L:time_of_day, enum to report the following values: Day 1; Dawn/Dusk 2; Night 4.

Please contact me if you have any questions.

Doug Dawson
douglassdawson@netscape.net
Toronto, Canada
November 6, 2006