

Differences between SWF and Windows applications

The SWF file exporter tries to make your SWF application work as close as possible as your Windows (stand-alone) application that you can see in action when you run it with the "Run Application" or "Run Frame" commands, but there are some differences between SWF and EXE applications. These differences are usually minor, but they can affect the way your program works. Here is the list of the main differences.

- **Background collision mask**

The way the background collision mask is handled is very different compared with the Windows runtime. In SWF applications, when you have a "No obstacle" object in front of an "Obstacle" or "Platform" background object, the "No obstacle" object will NOT have any effect on the collisions. Basically, the objects above it WILL have a collision detected, even if your topmost object is "No collision". The same limitation also occurs when you paste a "No obstacle" object in the background : it will not have any effect on the collisions.

We are aware of this limitation, and will do as much as we can to correct it in a future update.

- **String object, display of the string.** In the Windows runtime, the text of a string object can go over the vertical size of the object. In SWF applications, the text is clipped to the size of the object : you have to set the height of the String object large enough to contain all the text of the object.

- **String object, text height calculation**

Flash® Player calculates the height of a string of text differently than Windows. The result is that the height of the calculated text is higher than in Windows stand-alone applications. Therefore, vertically centered String objects will be displayed a little lower than in the Windows runtime. You will have to take this into account when positioning String objects in the frame.

- **INI and Hiscore objects**

Flash allows the application to save a small (under 100kb) data file on the user's machine. The files are saved in a folder on the user's machine, at the following address :
Users\UserName\AppData\Roaming\Macromedia\Flash Player\#SharedObjects

- **Edit box object.** When the object is in read-only mode, the text cannot be selected (with the mouse), this behaviour is different from the Windows runtime where it can be selected.

- **Sounds.** Adobe® Flash® Player plays MP3 sounds. Clickteam Fusion 2.5 automatically converts your WAV files into MP3 when you build a SWF file. If you change the sampling rate and bitrate of the MP3 converter, the SWF compiler might not accept your sounds in this format. In this case you will have an error message when building the application. If this occurs to you, change the sample rate of the MP3 converted to 44100 hz bitrate 128.

- **Sounds.** You cannot play external sample files (only internal samples).

- **MIDI music.** MIDI music is not supported by the Flash player.

- **Many properties** of the application and the frame available in Windows are not applicable to SWF applications. These properties will be grayed out when you turn your Windows application into a SWF File (by changing the Build type in the application's properties). After being grayed out, even if they are checked, they will have no effect. Example of properties without effect :

- "Keep display from previous frame".
- "Window style properties".
- "Window menu".
- "Change resolution mode".
- "Resize display to fill window size".
- "Machine Independent Speed".

- "etc."
- **Transitions.** Frames have no transitions and objects have only one type of transition (Fade).
- **Scrolling with background or active system box :** the differences between the way the objects are displayed make it impossible to have a scrolling background made of Background System Boxes. If you set their position in an event of the scrolling frame, the display will flicker.
- **Very large images** (bigger than 2880x2880) cannot be handled by the Adobe® Flash® Player. You will get an error message if you try to build an application with such images.
- **Keyboard and mouse events** do not occur at exactly the same time in the game loop. This might cause some problems in your application (like objects flashing when you set their position responding to a mouse click). Also right mouse clicks can't be intercepted.
- **Floating point calculations** are less precise than in Windows applications : usually a counter after a floating point calculation shows a lot of digits after the dot. To solve that problem, you should force the number of digits displayed after the dot in the counter object properties in the frame editor.
- **Due to internal encoding of floating point numbers**, comparison between floating point numbers may never be true. For example "Counter = 1.23" may not be true as the counter might be at 1.229999999. The solution here is to make a greater/smaller comparison with two conditions. Example "Counter>1.28" and "Counter<1.31". Note: this is true with the Windows runtime too! However calculations are more precise in the Windows runtime, so this problem may occur less often.
- **Mouse pointer.** ActionScript does not allow the retrieval of mouse coordinates out of the application's window area. This means that when the mouse cursor goes out of the frame, the last known coordinates will be the ones just before the mouse left the area. The mouse will then stay static until it comes back again in the frame.
- **There might be a display priority difference** between pasted objects and background objects on the same frame.
- **Layers with unlimited scrolling.** You must set the Wrap property of the first layer to ON so that the color of the background of the frame is displayed throughout the scrolling.
- **Frame rate.** You cannot change the frame rate at runtime (but you can change it in the properties).
- **Files.** You cannot load (or save) external files.