Introduction

There are various scenarios or different types of trials ActiveMARK can offer. This document is going to give a basic overview of the requirements for bundling games, in order to both be compatible with the wrapper but also to provide a high quality user experience.

Basic Requirements

1. General Wrapper Requirements

There is a set of core requirements that all games must comply with in order to be compatible with the wrapper. These are as follows:

Unprotected: Must be unprotected, and must not perform self-checks or self-reads (read data within the executable), or perform auto-updates (for example from the Internet).

Self-contained: Must not rely on CD, DVD, or external file or component not contained within its installer.

Win 32-code: Must a 32-bit Windows executable with a .exe extension, and must comply with standard Windows architecture.

Uncompressed: Must not be compressed or packed.

Intentional execution: The main executable must be called intentionally by the consumer (that is, it is not a device driver, Windows service or similar).

Macromedia Flash Games: Games which include Macromedia Flash must have the game's main SWF incorporated in the main executable.

Clean Exit: Must free all allocated resources (memory, COM, interfaces, etc.) before quitting.

2. How to handover your bundle game

Due to our wrapper and Fun Pass activation developers are requested to handover bundles in the following format.

A launcher executable must be provided for the bundle, our wrapper does not have a built in launcher capable of running multiple executables, so in order to provide an acceptable user experience this is required.

Critically there must also be dependency to the launcher added into the individual game exe's within the bundle. In order to be compatible with our desired user experience it's only possible for us to protect the launcher executable, with the individual game executables having to be left unprotected. This means of course the individual games must be tied to the game launcher; otherwise a user can just go into the game files and run the individual executables, which will not have protection. Thus it's essential that when providing bundle builds that you add dependency such that the individual games will not launch unless done through the launcher, and each exit when the launcher exits, or is forced to close by the wrapper at expiration.

Requirements:

- A launcher executable provided, from which you can run all desired games.
- Game executables must be dependent on the launcher:
    - They do not run without being done so through the launcher.
    - Exit when the wrapper force closes the launcher.

1. A bundle game has a launcher for two games with dependent executables, it has trial time left, and game 1 is running:

2. The trial time on the game expires, and the launcher closes:

3. The launcher is closed and as the executables are dependent on the launcher, the executable also closes.

4. After this expiration it should also not be possible to launch the individual games. If the user tries to run the launcher they will get a launcher process but it will only be a HTML page telling them that their game has expired.

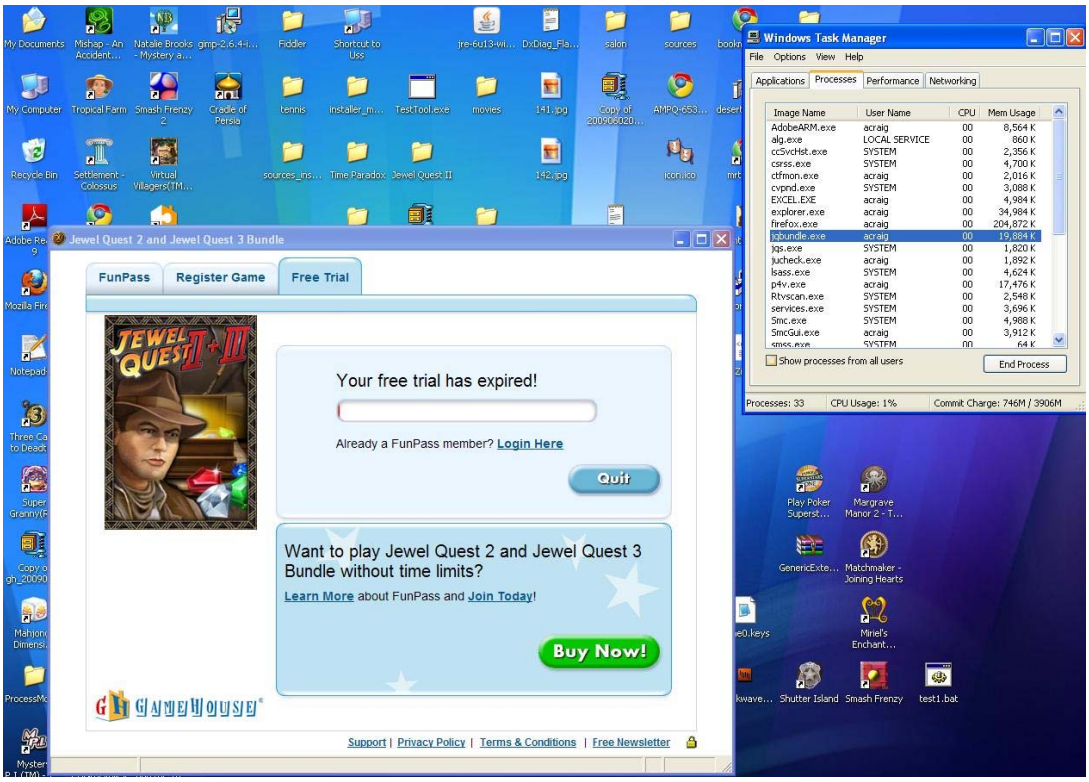
Summary

You can always ask us questions about whether a certain format is acceptable if you have problems you should contact us as far in advance of release as possible, and we will do what we can to help.

Examples

As you can see with this game, when a game from the bundle is ran, the launcher, which is what's protected by the wrapper, stays active.

Once the launcher is then forced to close by the wrapper, the dependency on the launcher means that the game exits as well.



After expiration, on running the launcher the user gets the expiration HTML page. It's critical that the dependency on the launcher means individual games cannot be ran at this point as well.