

Tip to select a GUI framework

Mitul Makadia



Tip to select a GUI framework

In 1970 when Apple launched their GUI based Macintosh computers, a new era in computer usability was started. Attractive GUI design is one of the key reasons of success for Microsoft Corporation.

Today there are vast choices to develop a GUI. Most of them operating provide implementation of a widely useful widget e.g. Button, Checkbox. Based on that, there are mainly two approaches to construct GUI.

1. Using native platform widgets

Frameworks on particular platform provide a variety of native widgets to construct GUI.

e.g. Visual Basic, VC++ on Microsoft windows, KDE on Linux, Cocoa on Mac and Java AWT.

In this approach, the constructed GUI resembles the OS look and feel.

2. Without using native platform widgets

Framework vendors create widget for each platform they support. An application can be ported on any of the supported platforms with a consistent look and feel. Currently, Java swing and Macromedia Flash are good choices to develop portable GUI. Applications developed using such frameworks require more memory compared to approach #1.

In recent years platform portability has become a major issue for GUI developers. In stark contrast, platform independent GUI development frameworks are slower and memory intensive. **Java SWT**, on the other hand, offers a solution which is both platform independent and consumes comparatively lesser memory. It exploits rich set of native widgets and creates implementation of only those widgets, not available on platform. **Eclipse IDE** is most widely accepted example of SWT.

❖ This table shows a comparison between different GUI development frameworks.

Technology Feature	Macromedia Flash	Java SWT	Java Swing	Microsoft VC++
Platform Independence	Yes	Yes(With platform specific library)	Yes	No
Well defined thread support	No	No	Yes	Yes
Execution speed	Fast	Faster	Slow	Faster
Programming Language	ECMA Script, partial OOP support	OOP	OOP	OOP

Memory leak	Yes	Yes	No	Yes
File handling	No	Yes	Yes	Yes
Type safety	Weak	Strong	Strong	Strong
Industry usage	Attractive & portable GUI development with less programming effort	Portable & Faster GUI development with native look and feel.	Complete portable GUI development with constant GUI across the platforms.	Faster GUI development with native look and feel with less programming effort.
Open source reusable components availability	Rare	Platform provides rich component set.	Platform provides rich component set.	Platform provides rich component set.
Maturity	Primitive	Primitive	Matured	Matured
Support available on internet	Little	Little	Abundant	Abundant