

Visual Perl/Tk

GUI development solution for
beginners and professionals



By **Felix Liberman**

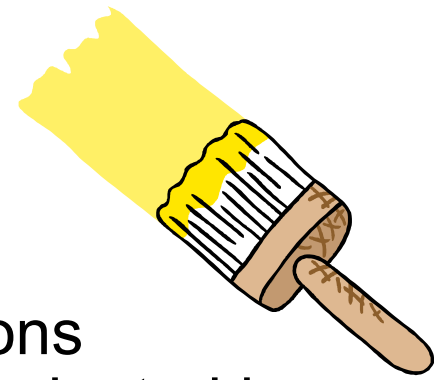
English version editor: **Uri Bruck**

Agenda

- Why Perl/Tk
 - Other languages/libraries
 - Comparison table
- Alternative visual tools for Perl/Tk
- What beginners need
- Why experienced programmer need it
- What VPTK offers
- Features that are not supported (yet)
- Demo



Why Perl/Tk
















We use Perl mostly for console or web applications development, while the GUI sector is already dominated by GTK, QT and tcl/tk, so why does Perl/Tk still remain a good alternative?

- We are familiar with Perl, so a new GUI environment ramp-up should not take much effort
- Perl has big advantage against other languages – it requires significantly less development and debug effort
- Perl/Tk runs on all GUI-compatible platforms, so it could be used when multiple platforms compatibility is critical
- We should consider that for really big GUI projects the Perl interpreter limitations could become a bottleneck, but contemporary HW/SW trends make this consideration increasingly outdated

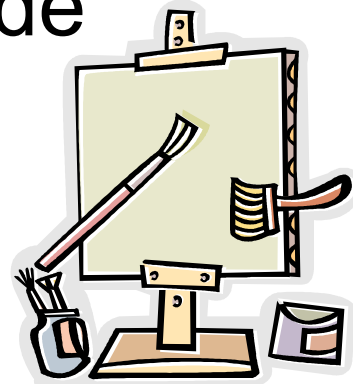


Why Perl/Tk (Comparison table)

	Perl/Tk	tcl/tk	Perl GTK	QT
Cross-platform			?	
Ready to run				
Easy ramp-up				
Development automation	?			?

Alternative visual tools for Perl/Tk

- If our choice is Perl/Tk we need a good tool for development automation
- There are many commercial solutions, but most are oriented to MS Win
- Official CPAN project “ZooZ” appears to be not quite ready for practical use
- “GLADE” and other open-source tools are too general, can’t produce instant code and require a specific platform or additional installations



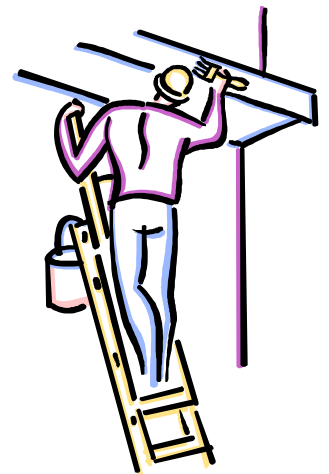
What beginners need

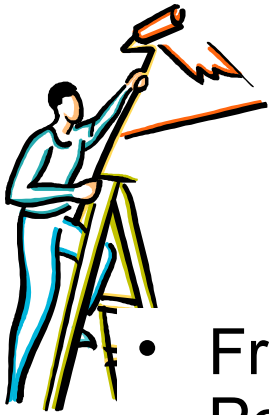


- When starting to learn a new GUI development package one wants to see what it offers and how it “kicks” in real life (not in a book)
- Beginners surely won’t be familiar with geometry managers and numerous widget options but need some practical way to learn them
- It’s easier for a beginner to “play” with his design interactively, without re-running the same program thousands of times
- For a beginner it’s important to see what generated code looks like – for future use of the same tricks in “manual mode”

Why experienced programmers need it

- Experienced programmers also need a visual environment for design automation:
 - For GUI sketching when working with customers
 - For quicker development of small projects that don't require special GUI tricks
 - For proof-of-concept experiments
 - For GUI project initial planning



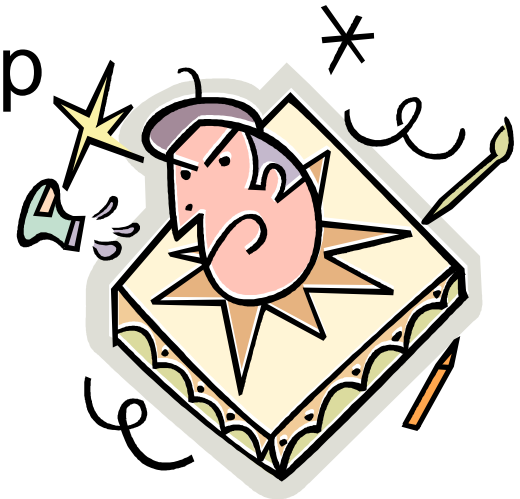


What VPTK offers

- Free, easy to install tool that works in every place where Perl/Tk installed
- Beginner-friendly context sensitive help based on Perl/Tk documentation
- User-side code support (callbacks & global variables)
- Two output formats: ready-to-run executable and sub-module code
- On-the-fly generated code preview and debug
- Geometry manager conflicts resolution (automatic)
- Undo/Redo
- Cut-'n'-Paste

Features that are not supported (yet)

- Non-static GUI tricks
- Balloons
- Lists/trees/tables contents
- Control over all widget's options
- Extended widgets set (like mega-widgets)
- Functional part before main loop
- Free input/output format
- Drag-'n'-drop interface



Demo



You can download for free VPTK (widget edition) here:

- <http://perltk.org/> => Scripts => General
- http://geocities.com/felixdaru/download/vptk_w.tgz

