Choosing the PEG+ or C/PEG Libraries

A Technical Whitepaper

James A. DeLisle
Kenneth G. Maxwell

Swell Software, Inc.

Revised 9-16-05

Summary

Embedded software developers evaluating Swell Software's C/PEG and PEG+ software libraries often entertain the following questions regarding the choice of the C/PEG or PEG+ library variants:

What is the difference between C/PEG and PEG+?
What features are missing in C/PEG that I would get with PEG+?
Which library is the best fit for my product development effort?

We will attempt to summarize these differences herein.

EC++ versus ANSI C

The first consideration is of course that PEG+ is written entirely in EC++, and C/PEG utilizes strict ANSI C. We do not attempt to state that either language is superior to the other. Rather, each has its own benefits. We do believe that EC++ is a natural fit for graphics and GUI programming. The use of inheritance allows the API set to be limited by re-using the same function names for many different classes. ANSI C is a bit more difficult to use for programming a GUI system such as C/PEG. This means that the API function list is more extensive, and it requires a bit more work for the developer to learn the API and learn to use it effectively.

However, the footprint for ANSI C programs tends to be smaller than C++ or EC++ programs. The footprint of C/PEG is roughly 60% that of PEG+. Much of this difference is due to a reduced feature set, but many compilers seem to produce smaller code and link smaller run-time libraries when only the ANSI C language is used. Also, the benefits of the EC++ language are not relevant if your programming team is not comfortable with the

C++ language and this should always be taken into consideration.

If the choice of CPU and compiler limit you to ANSI C, then the obvious choice is C/PEG. If your tools support both C and C++, you next should consider what is required of your GUI interface.

Target System Evaluation

There is a large overlap between targets that would be a good fit for C/PEG and those that are a good fit for PEG+. However, if you examine the "representative product" that each library is designed for, you might begin to see which product is a closer representation of your design. The following paragraphs describe in general terms the target devices for the C/PEG and PEG+ libraries.

C/PEG Target

C/PEG is designed primarily for smaller displays and limited CPU systems. These are very relative terms. By "small displays", we generally mean displays in the QVGA (320x240) pixel resolution or smaller. More specifically, if your display is too small to make practical use of overlapping top-level windows, this would fall into the small display category.

Because of the very small footprint (60K code space typical) for C/PEG, this library can be used on targets with very small ROM/FLASH resources, or even in DOS real-mode appliations. The C/PEG library is also a good fit for 8-bit controllers and all 16-bit microcontrollers.

C/PEG is designed for use on targets which support a maximum of 65 k colors (16 bpp color depths). The higher-level capabilities of PEG+ such as layered windows, alpha blending, and anti-aliased fonts do not fit well within the C/PEG target space. Monochrome, grayscale, and 256 color palette or packed pixel modes are the central target of the C/PEG library.

C/PEG applications typically use a simpler "Each screen takes up the entire display" application architecture. There is no support within C/PEG for overlapped windows with the visible portion of the background window continuously updating.

PEG+ Target

PEG+ is designed for more advanced systems with higher CPU horsepower and more advanced graphical capabilities. The minimum CPU recommendation for PEG+ is a 16-bit controller running at 25MHz, and advanced applications make full use of 32 and 64-bit architectures. External hardware-accelerated graphics controllers are fully leveraged for PEG+ scrolling and animation controls.

PEG+ supports alpha-blending, layering, run-time image decoders, and viewport maintenance allowing any number of overlapping windows. In practical use this usually requires a larger display size (QVGA or higher).

PEG+ can be tailored to fit smaller ROM and FLASH constraints for those who prefer to use the EC++ programming language, however the 120K typical code footprint can never be made as small as the footprint of C/PEG. PEG+ supports all target color depths from monochrome to 32-bit True-Color RGB+Alpha display modes.

PEG+ fully supports a many-window application architecture. There is no limitation on how many top-level windows can be visible, and background updating is fully supported. PEG+ applications can easily be scaled to very large projects with many screens and supporting widgets through efficient use of the class derivation features of the C++ programming language.

Feature Comparison

Head to head feature comparisons are difficult because many of the more advanced features of PEG+ can indeed be implemented using C/PEG. Many of the example programs provided with the library packages are very similar in appearance, and both libraries support all of the basic GUI control types needed to form the basis of a graphical interface. Indeed the more accurate question may be how much work is required by the developer as opposed to being a built-in feature of the library.

The following is however a brief list of features that are best described as automatically supported in PEG+ that are not available as built-in features of C/PEG:

- Viewports and overlapped top-level windows.
- 65K colors and higher
- Automatic horizontal and vertical scrolling of client areas.
- ZIP and UNZIP run-time compression/decompression.
- GIF, PNG, BMP, and JPG run-time image decoders.
- Charting widgets
- HMI widget set
- Docking MenuBar, ToolBar, status bar.
- Spreadsheet control
- HTML table style control
- Multi-line text edit control with auto-wrap and scrolling.
- Run-time string resource installation (i.e. Loadable language modules).

The PRESS and PEGasis add-on software libraries are based on PEG+, and require the PEG+ library.

Both PEG+ and C/PEG include, in very brief format, these and other gadget types:

- Windows (Panels)
- Text controls (edit field, transparent, etc)
- Various button types (radio, text, checkbox, bitmapped, decorated)
- Sliders and spin controls
- Scroll bars
- Complete set of application callable drawing primitives
- Icons

In addition, both PEG+ and C/PEG implement the basic event-driven architecture and support mouse, keyboard, touchscreen, and soft-key input types.

The PEG Development Toolkit (PDK) is fully compatible with both software libraries.

Conclusion

Advanced and very professional graphical interface applications can be developed using either the C/PEG or PEG+ libraries. If your main concern is the smallest possible footprint, C/PEG may be the best choice. If you want a larger built-in feature set and the advantages of a C++ API, PEG+ may be the best choice for your development. We hope that the above descriptions will help you determine the best fit for project.

For more information, please refer to the C/PEG and PEG+ product briefs and example programs.