

ReportLab - a Python reports library for direct PDF output

Andy Robinson
Robinson Analytics Ltd.

What is ReportLab?

Reportlab is a next-generation reporting package written in Python which directly outputs PDF documents. It lets you do just about anything with text, vector graphics and images. We hope it will be useful for

- o High-volume corporate reporting and database publishing
- o Dynamic PDF on the web
- o Adding a 'print engine' to Python applications
- o Shipping a report language with an application so that users can customize their own reports
- o Building documentation systems which index and archive output
- o A 'build system' for complex documents such as management accounts, statistical reports and scientific papers which need a lot of charts
- o Documenting software projects

What is the status?

The first well-formed PDF documents were produced over a year ago. The first application was a back end to the PIDDLE ("Plug In Drawing, Does Little Else") cross platform drawing library. I gave a demonstration at the Open Source conference in Monterey in August, and many Python developers have been helping to test, clean up and optimize the code. In October a new API was released and it has since been possible for people to produce reports with a minimum of programming.

Since then, development has been severely constrained by lack of time; however, there is now sufficient commercial interest to set up a development team and I expect things to develop very rapidly from February 1st.

PDFgen - exposing the PDF file format

ReportLab divides into a number of components. At the lowest level is the "pdfgen" library, which wraps up the details of the PDF file format. PDF is an indexed binary format with numerous cross-reference tables giving exact byte offsets into the file - you can't make PDF in Notepad! A 'canvas' class hides all this from the user and provides a wide variety of methods for positioning graphics and text, and exposes advanced PDF-specific capabilities such as compression and page transition effects.

PDFgen exposes all of the features needed for high-end publishing and vector graphics, including clipping, coordinate transformations and precise text metrics.

PLATYPUS - A Page Layout Model

The "next layer up", currently available in prototype, is codenamed PLATYPUS - "Page Layout and Typography Using Scripts". This is an object model which deals with page layout concepts such as frames on the page, and objects which know how to draw themselves and flow into frames. Out of this, one can rapidly build sophisticated reporting applications.

Wherever possible, formatting is driven by style sheets. For example, a Paragraph points to a ParagraphStyle in an external style sheet, allowing global changes to reports and consistency. Items such as tables, table cells and lines will also be placed in style sheets which can be re-used.

Two critical types of 'Drawable Objects' are tables and charts. It is hoped that an Open Source approach will lead to a growing library of sophisticated, easy-to-use objects which users can place in their reports.

While the current object model is basic, it already allows some sophisticated layout. We hope to evolve in the general direction of the XSL flow object model.

Applications

It is expected that people will use the ReportLab components to build specific reporting applications which know how to acquire and format data in a particular domain.

The first example is the "PythonPoint" application included in the package. This uses a simple XML languages to define frames on the page, text, bullets and images, and converts this into a finished PDF document without the user needing to write any Python code. This page was formatted as a PythonPoint slide.

The Demo

We will attempt to give a very brief overview of where ReportLab has got to, what it can do and how to get started using it.