

Portable Charting Language (PCL)

White Paper
Feb. 7th 2006

Thomas Chou
Chee-Seng Choong

1. Background

Charts and graphs have been indispensable tools since the beginning of time. They are used everywhere and by everyone. In the financial field, stock brokers analyze large amounts of financial data every day to determine what to buy and what to sell. In the medical field, clinicians look at patients' charts to determine the proper treatment. In the scientific and engineering fields, charts can be used to analyze massive amounts of scientific or performance data gathered over time revealing to their users trends and anomalies that they couldn't have discovered through other means.

Today, there are a plethora of charting tools out there each tailored for a specific purpose. But ultimately, all charting users are looking for the same thing and that is how to take a collection of data and produce a meaningful chart quickly and easily. Today's charting applications can't give you that. Spreadsheet applications such as Microsoft Excel are great for data entry but can be cumbersome when manipulating large amounts of data. Furthermore, it can't do batch processing - quickly generating charts from a series of input files. 3rd party charting libraries are highly configurable but require users to have a programming background to use them.

2. Goals

The purpose of PCL is to address the deficiencies that exist in today's charting tools described above. The language is designed to give users the ability to build charts customized to meet their individual needs without having them to understand the technical details behind writing a charting program.

2.1 Architecture Neutral and Portable:

PCL is an interpreted language and will execute on top of the Java Virtual Machine. Like Java, PCL is designed to run on any platform allowing users to produce charts in almost any environment.

2.2 Ease of use:

PCL provides the facilities to analyze, manipulate, and extrapolate the data by just writing a few lines of code. It hides the mundane details so that the user can concentrate on manipulating and analyzing the data itself. For example, the process of opening a file, creating an input stream, tokenizing, and parsing is simplified to just specifying the filename and the delimiter(s) used.

Function loops allow the programmer to take control of repetitive tasks such as deriving a new data series from existing data as well as enable batch processing of generating charts for multiple input files. Highlighting data points that exceed specific thresholds on the chart can be expressed in a little conditional statement. Basic tasks such as finding the min/mean/max/median can be expressed as a single keyword

3. Main Language Features

The following are a subset of the main features of the PCL language:

3.1 Data Types

The basic data types such as integer and float are supported for data values. Strings, which are mainly used as labels for data series, columns, and the chart's title, are also supported.

3.2 Basic Data Series Operations

The basic common operators such as +, -, *, / as well as mathematical functions like min, max, mean, median, are supported.

3.3 Program flow control

Basic flow control statements such as if-else, for, and while are available. These make it possible to analyze and selectively manipulate data. Users will also be allowed to create their own functions in addition to the predefined functions.

4. Sample Code

Below is one example of how a user could create a simple XY plot.

```
// Load data file to be plotted
chart.delimiters(",")
chart.load("c:\\data.csv")

// Control the size of the chart
chart.size.width = 100
chart.size.height = 100

// Identify your data columns
col[0].name = "xvalues column"
col[1].name = "yvalues column"

// Add a XY plot data series
dataseries[0].name = "series1"
dataseries[0].type = "xy plot"
dataseries[0].xvalues = col[0].data
dataseries[0].yvalues1 = col[1].data
```

```
// Plot the graph  
chart.plot()
```

`chart`, `dataseries`, and `col` are predefined keywords and objects that the programmer can use to manipulate the chart. There are also built in functions like `plot`, `min`, and `max` that users can use for data manipulation. Comments can be entered by using the `//` symbol.