**boost.format2 or** [**boost.io**](http://boost.io)

Neue format bibliothek, inkl. static format :D.

**Problem**

Das aktuelle Problem in C++ ist, dass die Ausgabe über I/O Stream gelöst werden, was mehrere Dinge auf einmal tut:

* format
* localization
* output

Was ich gerne hätte wäre eine Aufteilung, wobei der erste Schritt eine solide Format Bibliothek wäre, die besser als boost.format ist und std::localization unterstützt.  
D.h. es wäre ein format mit optionaler Lokalisation und dann die Daten direkt auf einen IO Device schreiben kann und damit keinen iostream braucht. Syntaktisch würde  
ich das wie python.format aussehen lassen.

**Features**

**Compile-Zeit aufbau des Formats**

Mit ein paar tricks lassen sich compile-zeit strings erstellen. Damit kann man die benötigte Größe für die Ausgabe errechnen (wenn die breite fix ist, also kein einfacher double) was hieße die Ausgabe ließe sich mit einer Allokation ODER gar auf dem Stack erstellen. Durch letzteres wäre das für Controller interessant.

So sollte der Spaß aussehen:

boost::format2(“{d} {i}”, 3.214, 42);  
//oder  
boost::format2(“{d} {i}”, std::localization(“en-US”), 3.214, 42);

Die Idee wäre python.format im Syntax zu imitieren.

Der Oberhammer wäre natürlich wenn das bidirektional funktionierte, d.h. folgendes ginge, aber das könnte sehr, sehr kompliziert werden:

tuple t = boost::format2(“{d} {i}”, “3.214 42”);

**Pros & Cons**

**Pro**

* Templates!
* Constexpr
* Weit benutzt
* Eventuell was für den Standard
* Verwendbar für Controller

**Contra**

* Templates!
* Relativ umfangreich
* Eventuell duale implementation

Einlesen bei boost.format