Introduction

With Rcpp attributes Rcpp modules (described in the Rcpp vignettes) it is easy to expose C++ classes and functions to R. This note describes how to use classes exported by modules in conjunction with functions exported using Rcpp attributes through the use of `RCPP_EXPOSED*` macros.

In the following snippets, a simple example is given of a simple class and a function that takes that class as an argument. The C++ function is exported using Rcpp attributes as follows:

```
#include

// [[Rcpp::export]]
void shout(std::string message) {
    Rcpp::Rcout << message << "!" << std::endl;
}
```

Now, calling this function from R is as easy as one can hope:

```
shout("Hello World")
```

```
Hello World!
```

C++ classes can be exported using Rcpp modules. The simple class `Echo` below has a `get()` method which returns the original constructor parameter.

```
#include
#include

using namespace Rcpp;

class Echo {
private:
      std::string message;
public:
      Echo(std::string message) : message(message) {}

      std::string get() {
        return message;
      }
};
```

This class can now be exposed to R by specifing the constructors and the methods that should be callable from R with

```
RCPP_MODULE(echo_module) {
      class_<Echo>("Echo")
      .constructor<std::string>()
      .method("get", &Echo::get)
      ;
};
```

Unfortunately, combining these two snippets as above creates a problem. The Rcpp attributes machinery that exports `shout()` will not be automagically aware of the `Echo` class. This will cause an error when the package is loaded by R as the required functionality that transforms the class

between a `SEXP` and a regular C++ object can't be loaded. The solution is simple: instruct the compiler to do so explicitly using the `RCPP_EXPOSED*` family of macros. In the current case it suffices to add

```
RCPP_EXPOSED_AS(Echo)
```

Now, constructing and using the class from R is again straightforward

```
echo <- new(Echo, "Hello World")
echo$get()
```

```
[1] "Hello World"
```

```
shout(echo$get())
```

```
Hello World!
```

The `RCPP_EXPOSED*` macros

Rcpp defines a number `RCPP_EXPOSED*` macros in `inst/include/Rcpp/macros/module.h`, the most important ones are

- `RCPP_EXPOSED_AS` which allows passing objects from R to C++. As seen above, this is needed when exported functions want to take a C++ object as argument. Other uses include methods and constructors of other Rcpp modules classes that take a C++ object as argument;
- `RCPP_EXPOSED_WRAP` which allows the other way around; This is needed when a exported function or method wants to return a C++ object;
- `RCPP_EXPOSED_CLASS` which allows both.