

C++ classes to extend and embed R: The Rcpp and RInside packages

Dirk Eddelbuettel¹

1. Debian Project

* Contact author: edd@debian.org

Keywords: R, C++, extensions, packages, embedding

This presentation discusses the `Rcpp` and `RInside` packages that can be used to extend R in high-performance computing settings by minimising the need for data transfer, translation or serialization. `Rcpp` is more generic and can be used to extend R with both custom code, or interfaces to existing libraries. `RInside` offers to take R directly into the user-driven problem domain by embedding it into a given application.

`Rcpp` provides a number of C++ classes that facilitate extending R with compiled code in C or C++. These classes provide a more natural and 'object-oriented' interface than the relatively low-level macros provided by R and documented in the *Writing R Extensions* manual.

We discuss the following classes

`RcppParams` accepts parameters from the calling R function via a named `list` which can contain components of type `double`, `int`, `string`, `bool`, as well as in C++ types for `Date` and `Datetime` object from R;

`RcppDate` accepts R `Date` objects; the class `RcppDateVector` provides a vectorised variant;

`RcppDatetime` accepts R `Datetime` objects; the class `RcppDatetimeVector` provides a vectorised variant; both operate at a microsecond resolution;

`RcppVector` accepts numeric R vector objects that can be of either type `integer` or `double`; the class `RcppVectorView` provides a lightweight view-only form;

`RcppMatrix` accepts numeric R matrix objects that can be of either type `integer` or `double`; the class `RcppMatrixView` provides a lightweight view-only form;

`RcppStringVector` accepts R vector objects of type `character`; the `RcppStringVectorView` provides a lightweight view-only form;

`RcppFrame` permits construction of `date.frame` objects at the C++ level; it supports any of the atomic types listed here plus `factor` types for the columns;

`RcppResultSet` permits construction of lists of objects to be returned to R; it can accept all of the types listed here plus a STL vectors and matrices, as well as `SEXP` object common to R.

and illustrate them with examples. We briefly mention more advanced components of `Rcpp` such as function callbacks.

The more recent `RInside` package builds on these classes. It refactors code from the `littler` scripting front-end to R by Horner and Eddelbuettel (2006, 2009) as C++ classes that make it easy to embed R in arbitrary C++ applications. We illustrate the use of these classes with examples.