# Exposing C++ classes and functions to Python

Exposing classes and functions defined in C++ libraries to Python is now possible in comma by creating C++/Python bindings with Boost.Python.

## Example

To illustrate this new capability, bindings for the C++ class `format` and its member function `size()` declared in `csv/format.h` have been defined in `python/comma/cpp_bindings/csv.cpp`:

```
// python/comma/cpp_bindings/csv.cpp
#include <boost/python.hpp>
#include <comma/csv/format.h>
BOOST_PYTHON_MODULE( csv )
{
    boost::python::class_< comma::csv::format >( "format",
boost::python::init< const std::string& >() )
        .def( "size", &comma::csv::format::size );
 // add other csv bindings here
}
```

and added to cmake:

```
# fragment of python/comma/cpp_bindings/CMakeLists.txt

add_cpp_module( csv csv.cpp comma_csv )
# add other modules here
```

Build comma with `BUILD_SHARED_LIBS=ON` and `BUILD_CPP_PYTHON_BINDINGS=ON`, then open a python interpreter and enter the following commands:

```
>>> import comma.cpp_bindings.csv as csv
>>> f = csv.format('d,2ub,s[5]')
>>> f.size()
15
```

The function `size()` outputs binary size corresponding to the format string that was passed to the format object `f` on construction.

## Under the hood

The bindings are placed inside a shared library and saved in the file `csv.so`, which is then installed in `comma/cpp_bindings` along with the other Python modules. On Ubuntu, it will usually be `/usr/local/lib/python2.7/site-packages/comma/cpp_bindings/csv.so`. Note that the name of the module used as a parameter for `BOOST_PYTHON_MODULE` macros has to match the name of the shared library declared in the cmake file, e.g. `csv` in the above example.

## Limitations

The bindings are exposed as a shared library and hence one is limited to building comma with shared libraries or ensuring that all required static libraries have been compiled with -fPIC. Attempting to link with static libraries without position independent code may cause linking to fail or link with shared libraries instead.