RBloomberg Manual

Ana Nelson

June 23, 2009

1 About RBloomberg

RBloomberg is an R package which handles fetching data from the Bloomberg financial data application. RBloomberg was written by Robert Sams, see the package README for additional contributors and acknowledgements. RBloomberg is released under a GPL open source license.

2 Installation and Requirements

2.1 Installation

RBloomberg will only work on a Bloomberg workstation, using the Desktop COM API. This version requires the rcom R package, which in turn depends on statconnDCOM which is a COM server that, while free, is not open source. Support for the Bloomberg Version 3 Java API is in development. When rcom is installed, it will give instructions on how to install statconnDCOM. If you install RBloomberg via

```
install.packages("RBloomberg", repos="http://R-Forge.R-project.org")
```

then rcom will automatically be installed for you if you don't have it.

2.2 Hello, World

Once you have RBloomberg installed, load the library just like any other via:

```
library(RBloomberg)
```

The first order of business is to connect to the Bloomberg data application, and store a reference to this connection object. You will use this in all subsequent calls:

```
conn <- blpConnect()
```

Then we can make a simple request to ensure that the connection is working:

```
blpGetData(conn, "RYA ID Equity", "NAME")
```

The result of running these three commands should be something like this:

```
> library(RBloomberg)
Loading required package: rcom
Loading required package: rscproxy
Loading required package: zoo

Attaching package: 'zoo'


        The following object(s) are masked from package:base :

         as.Date.numeric

Loading required package: bitops
Loading required package: RUnit
Contents of bbfields have been stored in .bbfields in the current workspace
Contents of bbfields.ovr have been stored in .ovr in the current workspace
> conn <- blpConnect()
> blpGetData(conn, "RYA ID Equity", "NAME")
                              NAME
RYA ID EQUITY RYANAIR HOLDINGS PLC
>
```

2.3 Unit Tests

Tests are live code examples that are tested for the expected output. They are useful to developers as a code quality tool, and they can also be very useful to users to help ensure everything is running smoothly and also as an extra source of reference material. Should you not be able to find the information you need in formal documentation (this is general advice, not just pertaining to RBloomberg), look for tests and study the syntax of examples there.

To ensure that everything is running smoothly, we recommend that you run the RUnit test suite:

```
testResults <- runTestSuite(allBloombergTests)
printTextProtocol(testResults)
```

The output of printTextProtocol should look like this:

```
RUNIT TEST PROTOCOL -- Tue Jun 23 15:06:37 2009
***********************************************
Number of test functions: 14
Number of errors: 0
Number of failures: 0


1 Test Suite :
All Tests - 14 test functions, 0 errors, 0 failures



Details
***************************
Test Suite: All Tests
Test function regexp: ^test.+
Test file regexp: Test.R$
Involved directory:
C:/DOCUME~1/nelsona/LOCALS~1/Temp/Rinst383694769/RBloomberg/runit-tests
---------------------------
Test file: C:/DOCUME~1/nelsona/LOCALS~1/Temp/Rinst383694769/RBloomberg/runit-tests/blpGetDataTest.R
test.basic: (2 checks) ... OK (1.66 seconds)
test.overrides: (9 checks) ... OK (2.14 seconds)
---------------------------
Test file: C:/DOCUME~1/nelsona/LOCALS~1/Temp/Rinst383694769/RBloomberg/runit-tests/blpToolsTest.R
test.category.name: (1 checks) ... OK (0.08 seconds)
test.data.type.for.list.of.fields: (1 checks) ... OK (0.03 seconds)
test.data.type.for.single.field: (1 checks) ... OK (0 seconds)
test.field.info.raises.error.on.invalid.mnemonic: (1 checks) ... OK (0.02 seconds)
test.field.name.for.list.of.fields: (1 checks) ... OK (0.03 seconds)
test.field.name.for.single.field: (1 checks) ... OK (0 seconds)
test.historical: (1 checks) ... OK (0 seconds)
test.is.power.of.two: (4 checks) ... OK (0 seconds)
test.static: (1 checks) ... OK (0 seconds)
test.what.i.override: (2 checks) ... OK (0.02 seconds)
test.what.overides.me: (2 checks) ... OK (0.03 seconds)
---------------------------
Test file: C:/DOCUME~1/nelsona/LOCALS~1/Temp/Rinst383694769/RBloomberg/runit-tests/rcomBloombergTest
test.bloomberg: (3 checks) ... OK (6.61 seconds)
```

In particular, take note of any errors or failures. Hopefully you will have none of either.