intro_recordr

Peter Slaughter

2016-05-18

Overview

The *recordr* package collects information about R script executions (aka runs). The captured information includes the files that were read and written by the R script, and details of the execution environment, such as the operating system version, R packages loaded, etc.

The recorded information for a script constitutes data provenance for the data products and analysis outputs (graphs, .csv files, etc) generated by a script execution, by providing information to describe how the data products were created.

Using recordr

Configuration

An initial package metadata template file has been copied to "~/.recordr/package_metadata_template.R" Please review the "recordr" package documentation section 'Configuring recordr' and then set the options parameters with values appropriate for your installation.

Recording a Script Execution

The *record()* method takes an R script as an argument and sources it, recording files that were read and written by R functions that are registered with *recordr*. It is not necessary to modify an R script in order to use *record*.

The following example runs a sample script that is included with the *recordr* package:

```r
library(recordr)
rc <- new("Recordr")
sampleScript <- system.file("extdata/EmCoverage.R", package="recordr")
runid <- record(rc, sampleScript , tag="first recordr run")
```

Information about the script execution is stored in the *recordr* cache (~/.recordr). *recordr* provides methods to search, view, modify and publish items stored in the cache. It is not recommended that files or directories be manually edited or deleted from the cache directories, with the exception of the items mentioned in this document.

Listing Script Executions

Script runs that have been recorded can be listed using the *listRuns()* method. The listing can be filtered by the tag value specified when a run was recorded. Runs can also be filtered by run start time, run end time, the text of error messages for a run and by a sequence number, which is an integer value assigned to each run to assist in easily specifying a particular run for listing, viewing or publishing.

In this example, all runs with a tag containing the string "first" are listed. Because recordr has only run once in this demo, only one run is listed:

```r
listRuns(rc, tag="first")
```

```
## Seq   Script                          Tag                   Start Time          End Time            Ru
## 13    /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 11:25:09 PDT 2016-05-18 11:25:1
## 12    /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 11:10:43 PDT 2016-05-18 11:10:4
## 11    /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 11:08:16 PDT 2016-05-18 11:08:1
## 10    /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 10:59:18 PDT 2016-05-18 10:59:1
## 9     /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 10:58:04 PDT 2016-05-18 10:58:0
## 8     /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 10:56:17 PDT NA
## 7     /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 10:55:55 PDT NA
## 6     /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 10:54:42 PDT NA
## 5     /Users/slaugh...a/EmCoverage.R first recordr run    2016-05-18 10:51:46 PDT NA
## 3     /Users/slaugh...a/EmCoverage.R first recordr run    2016-04-12 20:07:03 PDT 2016-04-12 20:07:0
## 2     /Users/slaugh...a/EmCoverage.R first recordr run    2016-04-12 19:52:53 PDT 2016-04-12 19:52:5
## 1     /Users/slaugh...a/EmCoverage.R first recordr run    2016-04-12 19:52:29 PDT 2016-04-12 19:52:2
```

If no search parameters are specified to *listRuns*, then all recorded runs are listed.

Preparing Metdata for a Run

The first time that recordr() is run, an initial metadata template file is copied to the file "~/.record/package_metadata_template. Each time that record() is called, a metdata file is created for the current run by using the template file as a starting point and generating an EML document, creating EML elements for the items in the template file. In addition 'otherEntity' element is created for each each data object that is created by the run and the R script that was run.

The metadata template file can be edited before a run, using the values you specify to affect the generated EML document.

If you are using Rstudio, click on File->Open File (Ctrl-O) and open ~/.recordr, then click on "package_metadata_template.R" in the File pane.

Currently only the items that are in the template file can be updated, and new elements cannot be added, so for example, the 'title', 'abstract' and 'creators' can be edited.

Recording An R Console Session

recordr can also collect information during an R console session using the *startRecord()* and *endRecord()* methods. When *startRecord()* is typed in the R console, information capture begins. Information will be captured for any function registered with *recordr*, while all other console input will not cause any information capture. Informaiton capture is terminated when *endRecord()* is entered in the console, and execution information is written to the *recordr* cache.

```r
startRecord(rc, tag="first console run")

df <- read.csv(file = system.file("./extdata/coverages_2001-2010.csv", package="recordr"))
endocladia_coverage <- df[df$final_classification=="endocladia muricata",]
myDir <- tempdir()
csvOutFile <- sprintf("%s/Endocladia_muricata.csv", myDir)
write.csv(endocladia_coverage, file = csvOutFile)

endRecord(rc)
```

The history of all statements typed during this recorded console session is saved in the *recordr* cache and will be included in the data package uploaded to a data repository when publishRun() is called.

Viewing Script Executions

More detailed information can be retrieved and viewed for a run or set of runs using the *viewRuns()* method.

```
viewRuns(rc, tag="console run", sections=c("details", "used", "generated"))
```

```
## list()
```

Information for all matching runs is retrieved and displayed, The output displayed by *viewRuns* is divided into the secions "details", "used" and "generated", which can be selectively displayed using the *sections* parameter.

Publishing

The *recordr* package uses the R package *dataone* to assist in packaging and publishing data generated by a script execution.

Uploading data to DataONE requires that a DataONE user identity be provided. For information about how this is done, please view the *dataone* package vignette *dataone-overview* by entering the R command `vignette("dataone-overview")` and reading the section *New Authentication Mechanism*.

The *dataone* package uses a configuration file to assist in how various operations are performed such as publishing a data package when the *publishRun()* method is called.

The first time that *record()* is run, an initial copy of the *dataone* configuration parameter file is copied to the directory '~/.dataone/sessionConfig.R'. *recordr* reads this file when *publishRuns()* is called, so this file can be edited before *publishRuns()* is called, for example, to control where the package is published to.

Open the file ~/.dataone/sessionConfig.R and make the following changes: - change the parameter dataone_env to "SANDBOX2" - change the parameter target_member_node to *urn:node:mnDemo2*

The metadata file that was created for the run can now be edited. Use the following calls to retrieve the metdata for the console session that was recorded and write it to a temporary file:

```
myMeta <- getMetadata(rc, seq=2)
tf <- tempfile(fileext=".xml")
writeLines(myMeta, tf)
```

At this point the metadata file can be edited.

Now save the file back into the *recordr* cache so that it will be included with the published package:

```
status <- putMetadata(rc, seq=2, metadata=tf, asText=F)
```

The parameter 'asText' tells *putMetdata()* that `metadata=tf` specifies a file to be read and not a character vector containing the metadata.

Now the run can be published, using the values that we specified for DataONE environment and memmber node, and usign the modified metadata document:

```
publishRun(rc, seq=2, quiet=F)
```

The uploaded data package can be viewed at 'http://mn-demo-2.test.dataone.org'