

RShell

Introduction

The RShell plugin for Taverna enables scientists to run R scripts within the Taverna environment. The processor uses the standard R interpreter (www.r-project.org) and therefore is able to run almost any R script. The R processor supports different data types, amongst others numeric, literal and image typed data.

Release notes of RShell 1.2.1

- **Support for R 2.7+**
- **Support for PDF output**
- **Support for syntax highlighting**
- **Easier to configure using less different port-types**
- Graphics (PNG) files can be returned to the Taverna client
- Text files can be send and returned to the Taverna client
- It supports multiple output ports
- It supports integers, doubles, strings, booleans and R-Expressions (REXP)
- Scripts can be loaded from file
- Each Rshell can connect to a different server
- Connection problem with Rserv is solved (was only visible with a Windows RServe)
- Connection can be kept alive, so one doesn't have to transfer all the data using workflows

Requirements

- Java 1.5 ([Java](#))
- Taverna 1.3 ([Taverna](#))
- R interpreter ([R-project](#))

The installation

The installation consists of two parts: installing the RShell plugin in Taverna and installing the RServe library for R. Since Taverna 1.5, RShell is part of the standard distribution of Taverna. This version of R perfectly works with R 2.3-2.6. If you use one of these versions, you can skip the Taverna plugin installation and can go further to the section [Installation of R and RServe].

A newer version of RShell with support for RShell 2.7+ is available for Taverna 1.7.1.

Installation of RShell 1.2 with support for R 2.7+ [for Taverna 1.7.1]

The newer RShell plugin can be installed using the Taverna plugin manager. Please follow the steps below to install this plugin:

1. Go to Tools->Plugin manager in the task bar.
2. Click on the "Find new plugins" button at the bottom of the dialog. In the next dialog click on the "Add Plugin Site".
3. Enter in the Site Name: `BioRange plugins`
4. Enter in the Site URL: `http://janus.cs.utwente.nl:8000/twiki/pub/BioRange/BioRangeSoftware/` The RShell plugin 1.2 will be listed in the plugin list.
5. Select this plugin using the checkbox and press the "Install" button. The RShell plugin is installed. HOWEVER, something more needs to be done. Since RShell is part of the standard Taverna distribution, the old plugin cannot be overwritten by the newer version. Taverna needs to be configured to use the newer version instead of the old one. Therefore, the Taverna configuration file needs to be modified slightly.
6. Close the Taverna application;
7. In Windows XP, the Taverna configuration file is located at: `%APPDATA%\Taverna-1.7.1\conf\current-profile-1.7.1.xml` (This is the same directory as: `c:\Documents and Settings\YOUR_USER_NAME\Application Data\Taverna-1.7.1\conf\current-profile-1.7.1.xml`) In unix based systems, like linux: `~/Taverna-1.7.1/conf/current-profile-1.7.1.xml`
8. Open this file in a text editor.
9. Search the line: `<artifact groupId="uk.org.mygrid.taverna" artifactId="taverna-contrib" version="1.7.1.0"/>`
10. And change the version number 1.7.1.0 into 1.7.1.1:
11. `<artifact groupId="uk.org.mygrid.taverna" artifactId="taverna-contrib" version="1.7.1.1"/>`

Now, Taverna is configured correctly to use the new RShell plugin. You can check this by adding an Rshell processor to the diagram, right clicking on the processor and selecting the "Configure RShel..." button. When you select to the "info" tab, it shows the version, which should be 1.2.

Next, the R installation needs to be configured to run as a server. See section [Installation of R and RServe] how to do this.

Installation [Taverna 1.3]

You have to download the `Rshell.jar` and the `JRClientRF_UT.jar` (see below). The first one is the Taverna plugin for running R scripts, the later is a library for establishing connections with an Rserve server for executing the R scripts. Our version of JRClient corrects a bug in the original version.

Put these two files in the `./lib` directory of your Taverna installation. Delete the old `JRClientXXX.jar` file in the `./lib` directory. Then, Taverna will automatically use the new JRClient file.

Downloads for Taverna

- [RShell.jar](#): The RShell
- [JRClientRF_UT.jar](#): Contains a correction in comparison to version JRclient-RF503.jar

See section [Installation of R and RServe] for the installation of R and RServe.

Installation of R and RServe

The RShell plugin communicates with the R interpreter using the RServe library. This library needs to be installed. The installation can directly be done using the R interpreter:

```
install.packages("Rserve")
```

The Rserve needs to be active when Taverna needs to execute R scripts. This can be done by using the following R commands:

```
library(Rserve)
```

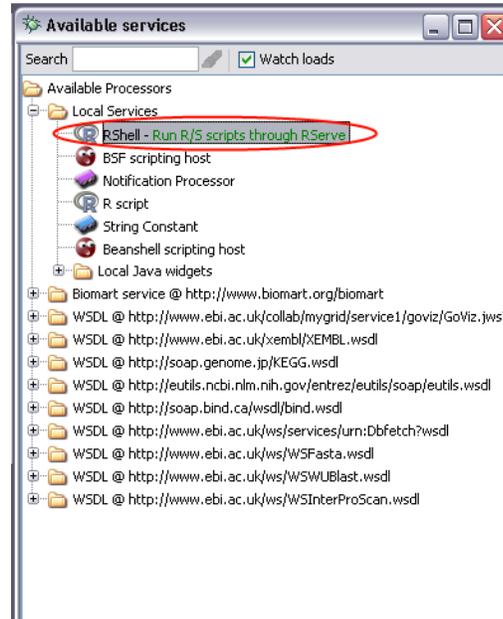
```
Rserve()
```

Or one can start R using the following command:

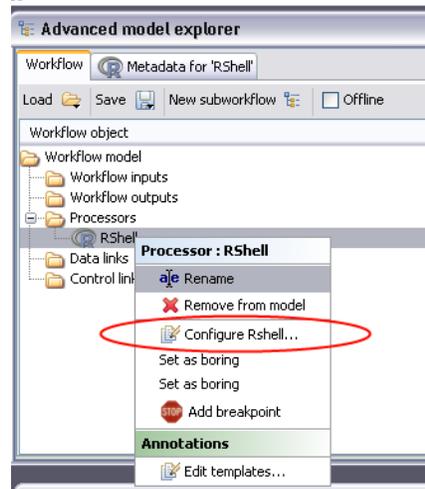
```
R --no-save -e library(Rserve);Rserve()
```

Creating a new Rshell process

The Rshell processes is available in the service explorer ("Available Services"). To create a new Rshell process, select the Rshell symbol and drag it to the "Advanced model explorer".

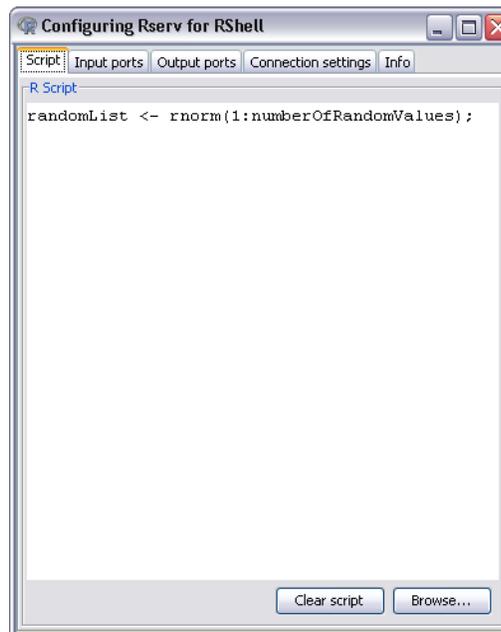


For setting the script, adding ports or modifying connection settings, the Rshell needs to be configured. In order to do this, right-click on the Rshell process in the "Advanced Model Explorer" and select "Configure Rshell...".



Rshell Script

The script-tab is used for typing the R-script to be executed by the Rshell process. The script can be typed manually, or can be read from an R-file (ending with ".R"). The input and output variables do not have to be declared here, since this is done in the input port tab and output port tab. In our example, the Rshell generates a list of random numbers. The length of the list is equal to numberOfRandomValues.



Creating ports

The Rshell has no ports by default. To add input ports and output ports, one has to select respectively the input port tab or output port tab. Each port has a name and is represented as a variable in the R script. Besides a name, every port has a type, which specifies the type of data it expects to be.

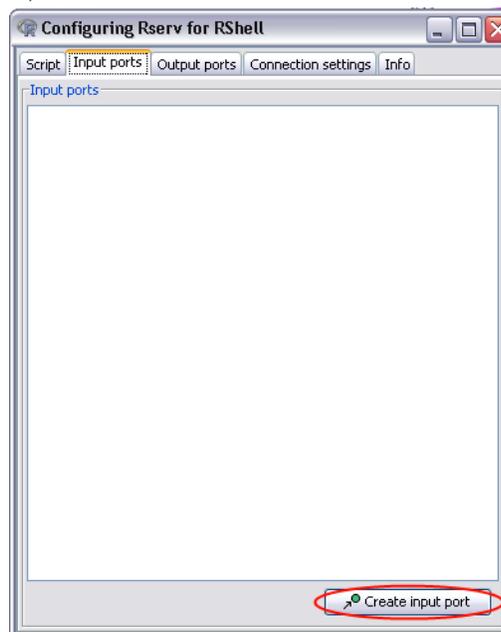
Port types

- **numeric** a floating point value
- **integer** a natural number
- **string** a string value
- **numeric[]** a list of numeric values
- **integer[]** a list of integers
- **string[]** a list of strings

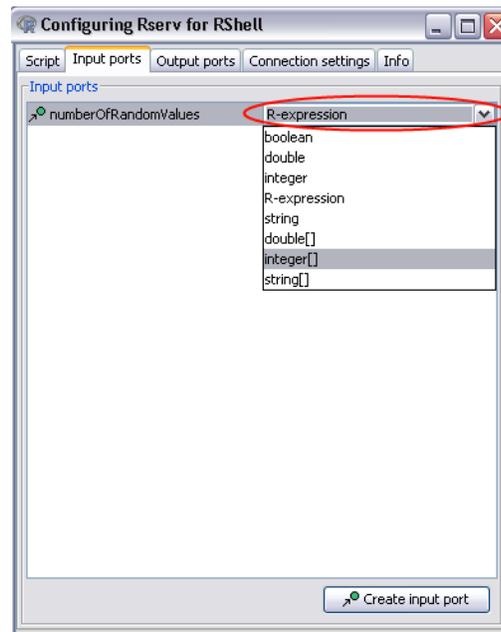
Input ports

Input ports are the port from which the Rshell can receive data coming from other processes in Taverna. Modifying the variable representing the input port in the R script has no influence on the input-port itself.

To add a new input port, click on the "Create input port" button. Now, a popup box will appear, where one can enter the port name. When the name is entered, the created port will appear in the port table.



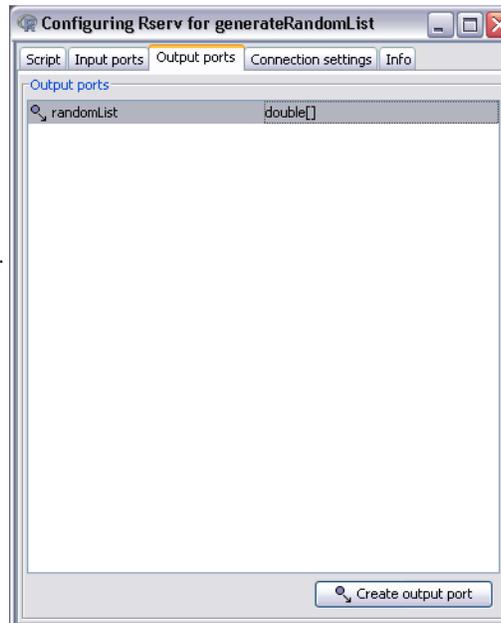
By default, a port is of type "R-expression". To modify the type, one can click on the current port type. Now, a drop-down list will appear. One can select one of the types in this drop-down list.



Output ports

Configuring connection to the Rserv

Output ports are used for passing data to other processes in Taverna. They represent the processors output. Creating and modifying the output port is done in the same way as input ports are created and modified. The name of the output port is also the variable representing the output port. Only the last assignment to the variable representing the output port is passed to the Taverna.



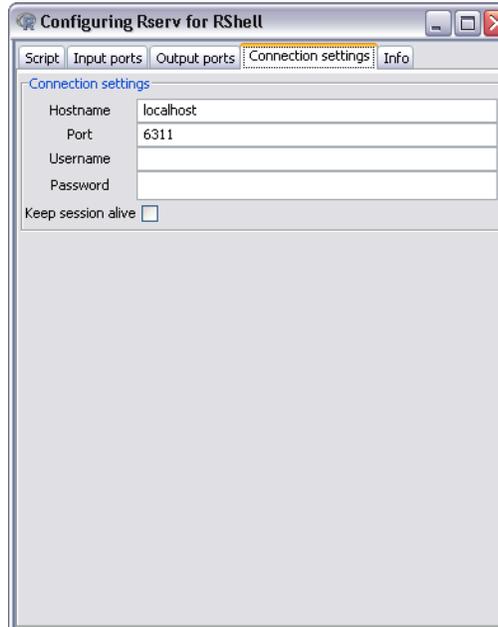
By default, the Rshell is configured for making connections with a Rserve on the same computer, in other words localhost. It can be the case that the Rserve is on another system. Then one has to change the connection settings.

The connection settings are:

- **Hostname** representing the url of the computer running Rserve
- **Port** this is the port number the client can connect the Rserve
- **Username** the username, if required, otherwise it needs to be blanc
- **Password** the password required for logging in, if not needed, leave it blanc

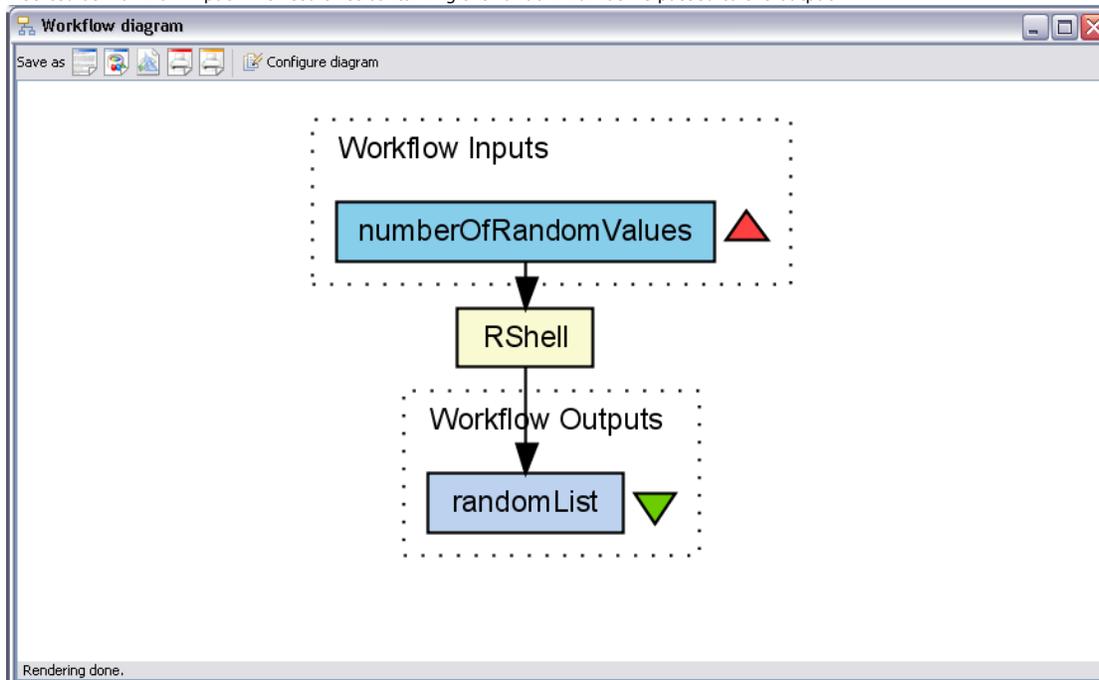
Besides these settings, there is an option called "Keep session alive". If this option is selected, then for all Rshell processes where this option is selected and which have the same connection settings, including username and password, share the same connection. The connection will be closed when all running workflows are finished.

This option is usefull to keep and to reuse variables stored at the Rserve. Normally when a connection is closed, all variables are thrown away or in other words, the session is closed. This can be used to prevent transferring huge amounts of data between Rshell processes. When a Rshell A process needs variables created in an other Rshell, named it B, then one needs to add a constraint between the processes A and B. This constraint prevents process A being executed before process B is finished.



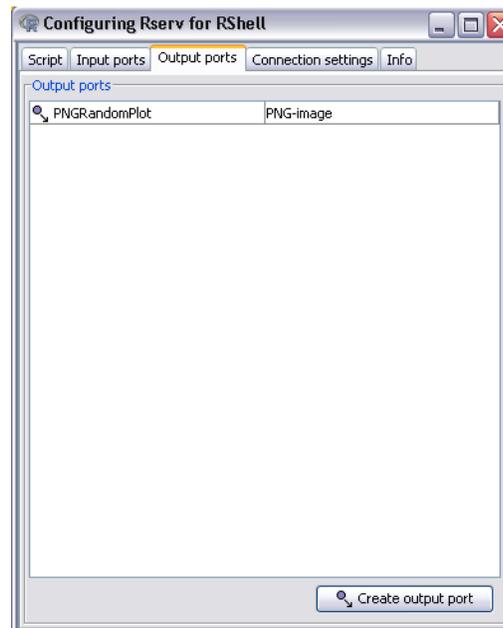
The result workflow

The figure below illustrates the created workflow. This workflow generates a list of N random numbers, where N is the number inserted as workflow input. The result list containing the random number is passed to the output.



Transferring images

Rshell supports transferring images to Taverna. This is done using normal ports. To do this, a port needs to be added with the name corresponding to the image (for example `PNGRandomPlot`).



When one wants to write to the image, the graphics device needs to be linked to png-device. This is done using the `png` command where the port name is the parameter of this command. When plotting is finished, the device needs to be closed, using the `dev.off()` command.

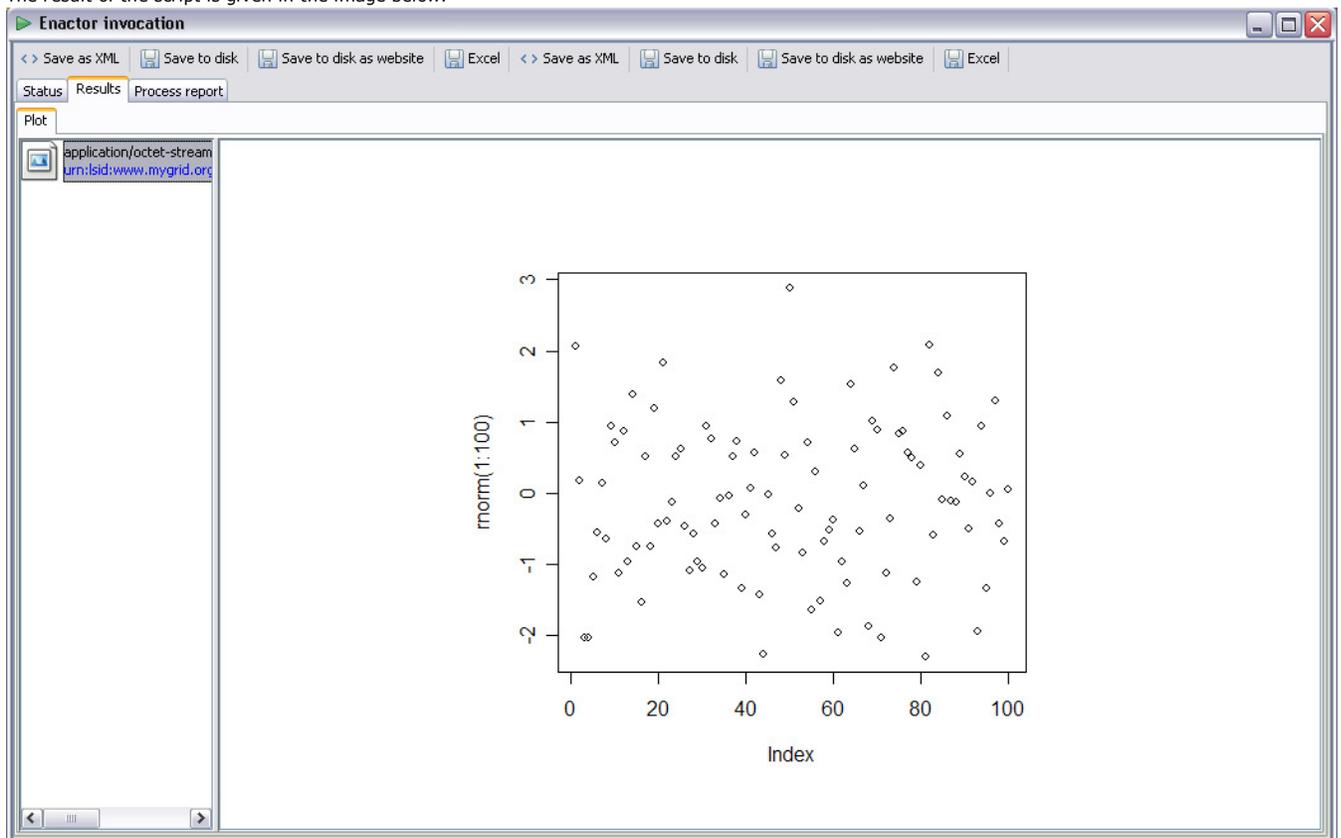
An example of the script is given below:

```
# Use the port name without quotes!  
png(PNGRandomPlot);  
plot(rnorm(1:100));  
dev.off()
```

When the port was configured to produce a pdf-file instead, the output script is configured in a similar way:

```
# Use the port name without quotes!  
pdf(PDFRandomPlot);  
plot(rnorm(1:100));  
dev.off()
```

The result of the script is given in the image below.



Transferring table files

It is possible to send (large) text files, such as tables, to the RShell plugin and to read them as tables. To do this, create an input port of type "Text-File". The input of this port will be automatically stored in a file called "myTable.txt". To read the table in R, use the following script:

```
# Use the port name without quotes!  
t <- read.table(myTable, header=TRUE)
```

Now the table can be used in the R script.

Example workflows

- **RTest.xml**: R test, shows how to use the R processor
- **PlotTest.xml**: Generates plots in PDF and PNG format
- **KeepAliveTest.xml**: Example how to use the keep-session-alive
- **ReadTextFile.xml**: Reads a table in R

Uninstall RShell 1.2 in Taverna 1.7.1

To uninstall RShell perform the following steps:

1. Remove RShell using the plugin manager (in the menubar: tools->plugin manager)
2. Select RShell and press the uninstall button
3. In Windows XP, the Taverna configuration file is located at: %APPDATA%\Taverna-1.7.1\conf\current-profile-1.7.1.xml
(This is the same directory as: c:\Documents and Settings\YOUR_USER_NAME\Application Data\Taverna-1.7.1\conf\current-profile-1.7.1.xml) In unix based systems, like linux: ~/.Taverna-1.7.1/conf/current-profile-1.7.1.xml
4. Open this file in a text editor.
5. Search the line: <artifact groupId="uk.org.mygrid.taverna" artifactId="taverna-contrib" version="1.7.1.1"/>
6. And change the version number 1.7.1.1 into 1.7.1.0:
7. <artifact groupId="uk.org.mygrid.taverna" artifactId="taverna-contrib" version="1.7.1.0"/>

Now, Rshell 1.2 is uninstalled. The old RShell plugin will be shown.

Source code

The source code can be downloaded from the SourceForge site using CVS. For more information, look at: <http://sourceforge.net/projects/rshell/>

-- IngoWassink - 08 Dec 2008

Revision: r1.9 - 11 Feb 2009 - 13:06 - [IngoWassink](#)

[BioRange](#) > [BioRangeSoftware](#) > RshellHelp
