

Chapter 88

Using the Adapter for Rserve

The Adapter for Rserve enables you to define a connection to Rserve for remotely executing R scripts to be used with WebFOCUS reports and charts as a summary (Compute) or virtual field.

The original Rserve paper is available at <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/Urbane.pdf>.

For additional information on Rserve, see <https://rforge.net/Rserve/index.html>.

Note: If you encounter any errors when using the Adapter for Rserve, consult the Session Log for details.

In this chapter:

- [Introduction to the Adapter for Rserve](#)
 - [Configuring the Adapter for Rserve](#)
 - [Creating a Synonym for an R Script](#)
-

Introduction to the Adapter for Rserve

Rserve is a TCP/IP server that allows users to run R scripts directly from WebFOCUS or the Reporting Server without the need to initialize R or link to an R library.

The adapter is available on UNIX, Linux, and Windows. On UNIX and Linux, Rserve allows parallel sessions and, therefore, the environment supports multiple users. On Windows, Rserve does not support parallel connections. Multiple sessions are shared, so the environment supports a single user at a time.

The installation of R and Rserve, and the creation of the model are done without server or WebFOCUS involvement, and are prerequisites for configuring and using the Adapter for Rserve.

Prerequisites

1. Install R and Rserve on a host computer that has access to the WebFOCUS Reporting Server. Install them in a location for which you have write access.
2. Create an R model that you will want to run from WebFOCUS.

3. Save the R model (known as serialization) in a user script folder accessible to Rserve, using the R save command or the saveRDS function. The name and location of this folder will be determined by the Rserve administrator.
4. Create a .csv file that has only those columns that will be used as independent variables in the model. Do not include the dependent variable.
5. Write an R script to read and run the model.

R Script Features

The R script that will read and run the model must conform to the following rules:

- ❑ The R script must read command line arguments, even though a command line interface is not used. In the following example, the command line arguments are saved to a variable named args.

```
args <- commandArgs(trailingOnly=TRUE)
```

- ❑ The first argument is the name of the file to be processed by the model. The file name and a temporary HOLD file are automatically generated by the server. The second argument is the name of the output file. For example:

```
input_file <- file.path(args[1])
output_file <- file.path(args[2])
```

- ❑ The result field name must match the output of the R script. For example, in the following syntax, the result field from the model is named Price:

```
colnames(results) <- c('Price')
write.csv(results, file=output_file, row.names=FALSE)
```

For a complete example, see [Sample Session: Creating a Synonym for an R Script and Running the Script](#) on page 2069.

Server and WebFOCUS Steps

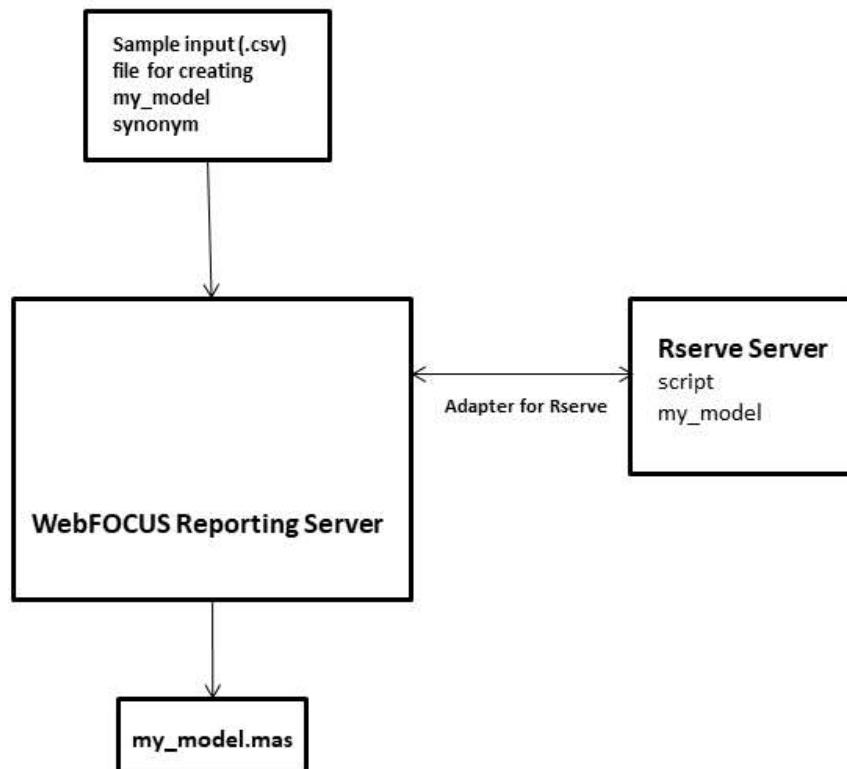
Once you have a model and the R script that runs it, you need a sample .csv file that contains a few rows of data for the independent variables used in the model. This file will be used to determine the data types and lengths of the independent variables when you create a synonym for the model.

Later, when you run the model from WebFOCUS, you will need a data file to run against that will provide values for the input variables to be passed to the RSERVE function. The function will return the result of running the model as output.

Therefore, the following steps must be completed in the server Reporting Server browser interface and WebFOCUS.

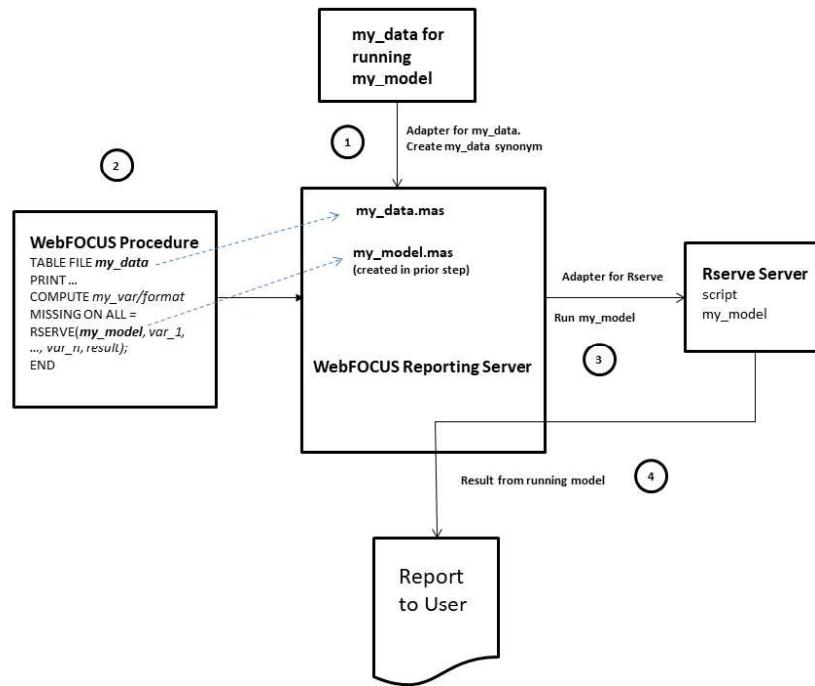
1. Configure the Adapter for Rserve, as described in [Configuring the Adapter for Rserve](#) on page 2066.
2. Upload the sample input file (.csv) to the server. This file is needed for creating the synonym for the model.
3. Create a synonym for the model, as described in [Creating a Synonym for an R Script](#) on page 2068.
4. Make sure an adapter has been configured for access to the data source against which you want to run the model.
5. Create a synonym for this data using the adapter configured in Step 4.
6. From WebFOCUS, create a report or chart that uses the RSERVE function in a COMPUTE, passes it the input values (independent variables), and retrieves the result, as described in [Sample Session: Creating a Synonym for an R Script and Running the Script](#) on page 2069.

The following diagram summarizes the process of preparing the model, named my_model, to be called in a WebFOCUS procedure.



[Configuring the Adapter for Rserve](#)

The following diagram summarizes the process of calling the model named my_model in a WebFOCUS procedure (against a data source supported by R).



Configuring the Adapter for Rserve

For information about prerequisites for running the adapter, right-click the adapter name on the Available list or a connection for the adapter on the Configured Adapters list, and click *Prerequisites*.

To configure the adapter, on the Get Data page of the Reporting Server Reporting Server browser interface, right-click *Rserve* on the Available list, and click *Configure*.

The Add Rserve to Configuration page opens.

Enter the following connection parameters.

Connection Name

Is a name for this connection.

Server

Specifies the Rserve host machine.

When specifying a host machine name other than localhost, ensure that remote access is enabled in the Rserve configuration file by adding the following:

```
remote enable
```

For information on creating the Rserve configuration file, refer to the Rserve documentation.

Rserve does not have to be installed on the Reporting Server host machine. However, it must be accessible to the Reporting Server.

Port

Is the port on which the Rserve host listens.

Security

There are three methods by which a user can be authenticated when connecting to Rserve:

- Explicit.** The user ID and password are explicitly specified for each connection and passed to Rserve, at connection time, for authentication.
If you select Explicit authentication, enter the Rserve user ID and password.
- Password Passthru.** The user ID and password received from the client application are passed to Rserve, at connection time, for authentication.
- Trusted.** The adapter connects to Rserve using the rules for an impersonated process that are relevant to the current operating system.

IBI_CLASSPATH

Are additional Java Class directories or full path jar names to be available for Java Services, each on a separate line. You must add the path to the location of the following jar files to IBI_CLASSPATH.

- On UNIX and Linux, the files REngine.jar and RServeEngine.jar are required. They are available on the Rserve web site, <https://www.rforge.net/Rserve/files/>.
- On Windows, the files REngine.jar and Rserve.jar are required. They are available in the Rserve zip file you can install from the RGui or download from the Rserve web site, <https://www.rforge.net/Rserve/files/>.

Click *Configure*.

If the configuration is successful, the following message displays.

```
Rserve successfully added to configuration
```

In addition, you can click *Test* on the context menu for the connection.

Creating a Synonym for an R Script

Creating a Synonym for an R Script

Each R script used with the Adapter for Rserve must have a synonym that describes the independent variables and dependent variable of the script. The Master File will contain the list of input (independent) variables and the output (dependent) variable. The Access File will contain information about the script and data files.

The synonym will be created using a sample file that contains only the fields that are input parameters for the script. A few rows of sample data are sufficient for the Adapter for Rserve to determine the appropriate data types and lengths of the parameters. The sample file must be a .csv file.

To create a synonym for an R script, right-click a connection for the Adapter for Rserve and click *Create metadata objects*. The Create Synonym for Rserve page opens, as shown in the following image, where the connection name is MyRserve.



Select or enter values for the following parameters.

R Script location on R server

Is the remote location, including script name and extension, for the R script file (.R). Leave blank if the R script is in an application folder accessible to the Reporting Server.

Note: In order to change a previously selected R script, you must delete the name of the file or the entire path before clicking the browse button.

Select file with sample input data for the R Script

Open the file picker (...) to select the application directory and file that contains the sample data for creating the synonym. Click *OK*.

Application

Open the file picker (...) to select the application that contains the R script. Select the R script file from the file picker and click *OK*.

Synonym Name

Enter a name for the resulting synonym, or accept the default name.

When you have finished entering the synonym creation parameters, click *Create Synonym* on the ribbon, as shown in the following image.



When you use the RSERVE function to run an R script from a WebFOCUS report or chart, the name and application location of this synonym will be passed as the first argument to the function. The other arguments will be the names of the independent variables and the dependent variable. For more information, see the *Using Functions* manual.

Sample Session: Creating a Synonym for an R Script and Running the Script

The following R script named wine_run_model.R predicts Bordeaux wine prices based on the average growing season temperature, the amount of rain during the harvest season, the amount of rain during the winter, and the age of the wine.

```
# filename: wine_run_model.r

args <- commandArgs(trailingOnly=TRUE)
input_file <- file.path(args[1])
output_file <- file.path(args[2])

wine_test <- read.csv(input_file)

wine_model <- readRDS('/prediction/wine_model.rds')

results <- predict(wine_model, newdata = wine_test)

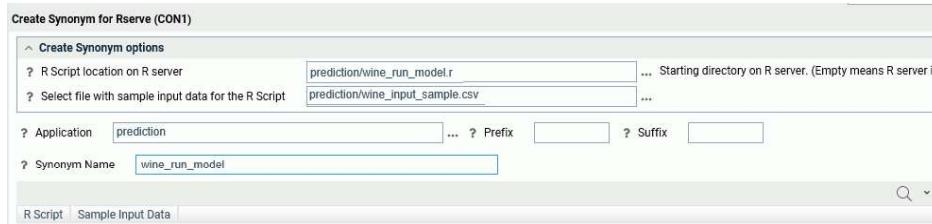
results <- as.data.frame(results)
colnames(results) <- c('Price')
write.csv(results, file=output_file, row.names=FALSE)
```

The following sample data file named wine_input_sample.csv contains the names and sample values for the independent variables used in this model.

```
"AGST", "HarvestRain", "WinterRain", "Age"
16.1667, 122, 717, 4
16, 74, 578, 3
```

Creating a Synonym for an R Script

The synonym creation page for this script is shown in the following image.



Clicking Create Synonym on the ribbon generates the wine_run_model synonym. The Master File, wine_run_model.mas, describes the independent (input) variables and the dependent (output) variable, as shown below:

```
FILENAME=WINE_RUN_MODEL, SUFFIX=RSERVE , $  
SEGMENT=INPUT_DATA, SEGTYPE=S0, $  
FIELDNAME=AGST, ALIAS=AGST, USAGE=D9.4, ACTUAL=STRING,  
MISSING=ON,  
TITLE='AGST', $  
FIELDNAME=HARVESTRAIN, ALIAS=HarvestRain, USAGE=I11, ACTUAL=STRING,  
MISSING=ON,  
TITLE='HarvestRain', $  
FIELDNAME=WINTERRAIN, ALIAS=WinterRain, USAGE=I11, ACTUAL=STRING,  
MISSING=ON,  
TITLE='WinterRain', $  
FIELDNAME=AGE, ALIAS=Age, USAGE=I11, ACTUAL=STRING,  
MISSING=ON,  
TITLE='Age', $  
SEGMENT=OUTPUT_DATA, SEGTYPE=U, PARENT=INPUT_DATA, $  
FIELDNAME=PRICE, ALIAS=Price, USAGE=D18.14, ACTUAL=STRING,  
MISSING=ON,  
TITLE='Price', $
```

The Access File, wine_run_model.acx, describes the names and locations of the R script and the sample data file, as shown below.

```
SEGNAME=INPUT_DATA,  
CONNECTION=MyRserve,  
R_SCRIPT=/prediction/wine_run_model.r,  
R_SCRIPT_LOCATION=WFRS,  
R_INPUT_SAMPLE_DAT=prediction/wine_input_sample.csv, $
```

Now that the synonym has been created for the model, the model will be used to run against the following data file named wine_forecast.csv.

```
Year,Price,WinterRain,AGST,HarvestRain,Age,FrancePop
1952,7.495,600,17.1167,160,31,43183.569
1953,8.0393,690,16.7333,80,30,43495.03
1955,7.6858,502,17.15,130,28,44217.857
1957,6.9845,420,16.1333,110,26,45152.252
1958,6.7772,582,16.4167,187,25,45653.805
1959,8.0757,485,17.4833,187,24,46128.638
1960,6.5188,763,16.4167,290,23,46583.995
1961,8.4937,830,17.3333,38,22,47128.005
1962,7.388,697,16.3,52,21,48088.673
1963,6.7127,608,15.7167,155,20,48798.99
1964,7.3094,402,17.2667,96,19,49356.943
1965,6.2518,602,15.3667,267,18,49801.821
1966,7.7443,819,16.5333,86,17,50254.966
1967,6.8398,714,16.2333,118,16,50650.406
1968,6.2435,610,16.2,292,15,51034.413
1969,6.3459,575,16.55,244,14,51470.276
1970,7.5883,622,16.6667,89,13,51918.389
1971,7.1934,551,16.7667,112,12,52431.647
1972,6.2049,536,14.9833,158,11,52894.183
1973,6.6367,376,17.0667,123,10,53332.805
1974,6.2941,574,16.3,184,9,53689.61
1975,7.292,572,16.95,171,8,53955.042
1976,7.1211,418,17.65,247,7,54159.049
1977,6.2587,821,15.5833,87,6,54378.362
1978,7.186,763,15.8167,51,5,54602.193
```

The data file can be any type of file that R can read. In this case it is another .csv file. This file needs a synonym in order to be used in a report request.

The following image shows the synonym creation page for wine_forecast.csv using the Adapter for Delimited Files.

Creating a Synonym for an R Script

The following is the generated Master File, wine_forecast.mas.

```
FILENAME=WINE_FORECAST, SUFFIX=DFIX      , CODEPAGE=1252,
      DATASET=prediction/wine_forecast.csv, $
SEGMENT=WINE_FORECAST, SEGTYPE=S0, $
    FIELDNAME=YEAR1, ALIAS=Year, USAGE=I6, ACTUAL=A5V,
      MISSING=ON,          TITLE='Year', $
    FIELDNAME=PRICE, ALIAS=Price, USAGE=D8.4, ACTUAL=A7V,
      MISSING=ON,          TITLE='Price', $
    FIELDNAME=WINTERRAIN, ALIAS=WinterRain, USAGE=I5, ACTUAL=A3V,
      MISSING=ON,          TITLE='WinterRain', $
    FIELDNAME=AGST, ALIAS=AGST, USAGE=D9.4, ACTUAL=A8V,
      MISSING=ON,          TITLE='AGST', $
    FIELDNAME=HARVESTRAIN, ALIAS=HarvestRain, USAGE=I5, ACTUAL=A3V,
      MISSING=ON,          TITLE='HarvestRain', $
    FIELDNAME=AGE, ALIAS=Age, USAGE=I4, ACTUAL=A2V,           MISSING=ON,
      TITLE='Age', $
    FIELDNAME=FRANCEPOP, ALIAS=FrancePop, USAGE=D11.3, ACTUAL=A11V,
      MISSING=ON,          TITLE='FrancePop', $
```

The following is the generated Access File, wine_forecast.acx.

```
SEGNAME=WINE_FORECAST,     DELIMITER=',',     ENCLOSURE=",",   HEADER=YES,
CDN=COMMAS_DOT,           CONNECTION=<local>, $
```

The following request, wine_forecast_price_report.fex, uses the RSERVE bulit-in function to run the script and return a report.

```
-*wine_forecast_price_report.fex
TABLE FILE PREDICTION/WINE_FORECAST
PRINT
  YEAR
  WINTERRAIN
  AGST
  HARVESTRAIN
  AGE

  COMPUTE PREDICTED_PRICE/D18.2 MISSING ON ALL=
    RSERVE(prediction/wine_run_model, AGST, HARVESTRAIN, WINTERRAIN, AGE, Price);
      AS 'Predicted,Price'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

The output is shown in the following image.

<u>Year</u>	<u>WinterRain</u>	<u>AGST</u>	<u>HarvestRain</u>	<u>Age</u>	<u>Predicted Price</u>
1952	600	17.1167	160	31	7.72
1953	690	16.7333	80	30	7.87
1955	502	17.1500	130	28	7.68
1957	420	16.1333	110	26	7.00
1958	582	16.4167	187	25	7.02
1959	485	17.4833	187	24	7.54
1960	763	16.4167	290	23	6.76
1961	830	17.3333	38	22	8.36
1962	697	16.3000	52	21	7.51
1963	608	15.7167	155	20	6.63
1964	402	17.2667	96	19	7.56
1965	602	15.3667	267	18	5.92
1966	819	16.5333	86	17	7.56
1967	714	16.2333	118	16	7.11
1968	610	16.2000	292	15	6.26
1969	575	16.5500	244	14	6.60
1970	622	16.6667	89	13	7.32
1971	551	16.7667	112	12	7.19
1972	536	14.9833	158	11	5.88
1973	376	17.0667	123	10	7.09
1974	574	16.3000	184	9	6.57
1975	572	16.9500	171	8	6.99
1976	418	17.6500	247	7	6.92
1977	821	15.5833	87	6	6.71
1978	763	15.8167	51	5	6.91

Creating a Synonym for an R Script