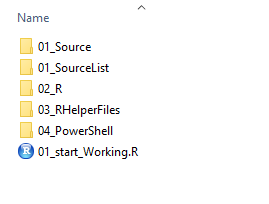
Installing R from scratch and creating your favorite IDE setup is especially useful when making fresh installation or when you are developing and testing out different versions.

This blogpost will guide you through some essential steps (hopefully, there will not be many) on how to download the desired [R engine](https://cran.r-project.org/), desired R GUI – in this case [RStudio](https://www.rstudio.com/), and how to prepare the additional packages with some custom helper functions to be used in the client set-up / environment. And mostly, using [PowerShell](https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-6) script.



Test folder for this new R Environment will be: *C:\DataTK\99\_REnv\01\_Source\*.  And the rest of the folder structure will be:



Folder structure is completely arbitrary and can be changed, accordingly.

**1. Downloading the RStudio and R**

All the programs will be installed with predefined paths (Please note, this path might vary on your client machine):

* RStudio ->  *c:\Program Files\RStudio*
* R Engine *-> c:\Program Files\R\R-3.5.1*

Both paths can be different on your machine. In the folder structure, I will set my folder pointing to 01\_Source sub-folder, as shown in ps script.

$dir = "C:\DataTK\99\_REnv\01\_Source\"

Set-Location $dir

## Download RSTudio for Windows machine

# Version of RStudio is deliberatly set to specific version

# so that code is repeatable and always returns same results

$urlRStudio = "https://download1.rstudio.org/RStudio-1.1.463.exe"

$outputRStudio = "$dir\RStudio.exe"

$wcRStudio = New-Object System.Net.WebClient

$wcRStudio.DownloadFile($urlRStudio, $outputRStudio) # $PSScriptRoot

Write-Output "Download Completed"

## Download R engine for Windows machine

$urlR = "https://cran.r-project.org/bin/windows/base/R-3.5.2-win.exe"

$outputR = "$dir\R-win.exe"

$wcR = New-Object System.Net.WebClient

$wcR.DownloadFile($urlR, $outputR)

Write-Output "Download completed"

## Installing R / RStudio on desired Path

## Silent install

$dirRStudio = $dir + "RStudio.exe"

$dirR = $dir + "R-win.exe"

Start-Process -FilePath $dirRStudio -ArgumentList "/S /v/qn"

Start-Process -FilePath $dirR -ArgumentList "/S /v/qn"

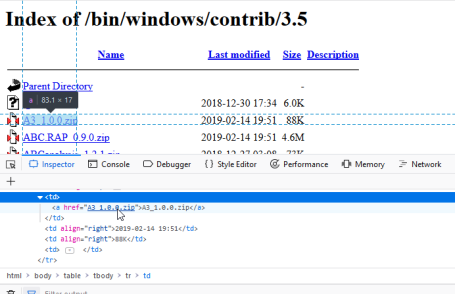
Now that we have the R engine and R Studio installed, you need to repeat the process for downloading the R Packages. In same manner, I will start downloading the specific R packages.

**2. Downloading the R packages**

For the brevity of this post, I will only download couple of R packages from CRAN repository, but this list is indefinite.

There are ways many ways to retrieve the CRAN packages for particular R version using powershell. I will just demonstrate this by using Invoke-WebRequest cmdlet.

Pointing your cmdlet to URL: <https://cran.r-project.org/bin/windows/contrib/3.5>  where  list of all packages for this version is available. But first we need to extract the HTML tag where information is stored. Since the URL stores data in a table, we have to navigate to following tag: *html>body>table>tbody>tr>td>a* where the file name is presented.



Packages names is retrieved by:

2019-02-17 07_45_48-Window

$ListRPackages= Invoke-WebRequest -Uri "https://cran.r-project.org/bin/windows/contrib/3.5"

$pack = ($ListRPackages.ParsedHtml.getElementsByTagName('a')).outerText

If you have the list of needed packages listed in a txt file, you can read the package names from file and iterate through the webpage and download the files:

$ListPackageLocation = "C:\DataTK\99\_REnv\01\_SourceList\packages.txt"

$PackList = Get-Content -Path $ListPackageLocation

$dir = "C:\DataTK\99\_REnv\01\_Source\"

ForEach ($Name in $PackList)

{

$UrlRoot = "https://cran.r-project.org/bin/windows/contrib/3.5/"

$url = $UrlRoot + $Name

$FileName = $dir +'\' + $Name

$PackagesOut = New-Object System.Net.WebClient

$PackagesOut.DownloadFile($url, $FileName)

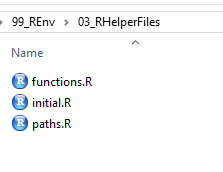
Write-Output "Download Completed"

}

Now that we have all the packages downloaded and programs installed, we can move to R.

**3. Setting up the R Environment**

In the folder structure, there is a folder including the helper files:



**Paths.R**

In this file all the paths are typed and later used in any other file. Simply the folder structure is described:

sourcePath = "c:\\DataTK\\R\_packages\\01\_Source"

sourcePackagePath = "c:\\DataTK\\R\_packages\\01\_Sourcelist"

libPath = "C:\\DataTK\\R\_Packages\\02\_R"

wdPath = "C:\\DataTK\\R\_Packages"

**Functions.R**

This file includes all the functions lists in one place, mainly for sharing or creating shared projects. In  this case, just two functions, one for checking and installing missing packages, read from the folder structures (that were previously downloaded using powershell).

# Function for sum of squares for two input integers

sum\_squares <- function(x,y) {

x^2 + y^2

}

# Function for package installation with check for existing packages

function\_install\_4 <- function(df\_name) {

for (i in 1:nrow(df\_name)){

if (df\_name[i,2] %in% rownames(installed.packages(lib.loc=libPath))){

#print(df\_name[i,2])

print(paste0("Package ",df\_name[i,2], " already installed."))

} else {

install.packages(df\_name[i,1], type="source", repos=NULL, lib=libPath)

}

}

}

**Intial.R**

This file wraps all the helper files in one place and invokes all the functions from packages and paths:

# Loading files with function lists and Paths

source(file="C:\\DataTK\\R\_Packages\\paths.R")

source(file="C:\\DataTK\\R\_Packages\\functions.R")

#updating the list of packages

setwd(sourcePackagePath)

listPackages <- data.frame(read.csv("packages.txt", header=FALSE))

names(listPackages)[1] <- "name"

#just names of the packages

temp <- strsplit(as.character(listPackages$name),"\_")

temp <- data.frame(library=matrix(unlist(temp), ncol=2, byrow=TRUE)[,1])

listPackages<- cbind(name=listPackages, library=temp$library)

#installing the missing packages

setwd(sourcePath)

function\_install\_4(listPackages)

library(dplyr, lib.loc=libPath)

library(ggplot2, lib.loc=libPath)

library(knitr, lib.loc=libPath)

**4. Start using R**

Finally, every new R file or projects needs to have a single line included:

#initialize

source(file="C:\\DataTK\\R\_Packages\\initial.R")

And this will load all the settings, all the packages and make sure the environment downloaded are correctly installed.

As always, complete code is available on [Github](https://github.com/tomaztk/Install_setup_R_using_Powershell).

Happy coding and happy Rrrrring