Are you a Microsoft Windows R user? Does your Windows username include a space? Like Firstname Lastname. Then you might occassionally run into issues installing packages due to spaces.

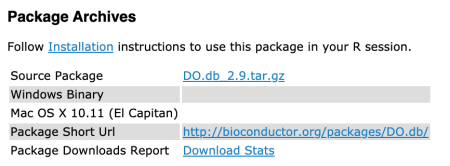
**Solutions**

You could either re-install Windows with a username that has no spaces such as Lastname [1](https://feeds.feedburner.com/2019/09/18/windows-user-space-issues-with-installing-r-packages/#fn1), but that’s probably not an easy option. Or you can:

* Edit your TMP and TEMP environment variables to a location with no spaces, like C:\TEMP following instructions like [these ones](https://www.howtogeek.com/285710/how-to-move-windows-temporary-folders-to-another-drive/).
* Preferably install R at a location with no spaces, like C:\R, instead of the default C:\Program Files.

**Backstory**

A co-worker wanted to install the [*clusterprofiler*](https://bioconductor.org/packages/3.9/clusterprofiler) Bioconductor package which depends on the [*DO.db*](https://bioconductor.org/packages/3.9/DO.db) Bioconductor package. This co-worker uses a Windows machine that has a username with a space. Let’s say it was me with Leo Collado to keep them anonymous. The [*DO.db*](https://bioconductor.org/packages/3.9/DO.db) is only available as a “Source” package with no Windows binary as you can see [here](http://bioconductor.org/packages/release/data/annotation/html/DO.db.html).



This means that R has to:

1. download a tar.gz file,
2. uncompress it,
3. and then install it.

In particular, we are talking about [DO.db\_2.9.tar.gz](http://bioconductor.org/packages/release/data/annotation/src/contrib/DO.db_2.9.tar.gz) in this case.

The installation instructions for [*DO.db*](https://bioconductor.org/packages/3.9/DO.db) are:

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("DO.db")

**Uncompressing**

Internally, BiocManager::install() ends up using utils::install.packages(). The first step, downloading, works well. Uncompressing a file in this scenario fails. Why?

> BiocManager::install('DO.db', lib = 'C:/R/R-3.6.0/library')

Bioconductor version 3.9 (BiocManager 1.30.4), R 3.6.0 (2019-04-26)

Installing package(s) 'BiocVersion', 'DO.db'

## removed output

trying URL '<https://bioconductor.org/packages/3.9/data/annotation/src/contrib/DO.db_2.9.tar.gz>'

Content type 'application/x-gzip' length 1769978 bytes (1.7 MB)

downloaded 1.7 MB

Error in untar2(tarfile, files, list, exdir, restore\_times) :

incomplete block on file

The downloaded source packages are in

‘C:\Users\Leo Collado\AppData\Local\Temp\RtmpqiBJ53\downloaded\_packages’

We were told to try installing R at a location with no spaces, so by this point, R was installed at C:\R\R-3.6.0\, hence the lib specification you see above, though it’s irrelevant for these errors.

Uncompressing the tar.gz file is done by utils::untar(). If you look at the code for utils::untar() you’ll see:

## The function definition of utils::untar

function (tarfile, files = NULL, list = FALSE, exdir = ".", compressed = NA,

extras = NULL, verbose = FALSE, restore\_times = TRUE,

support\_old\_tars = Sys.getenv("R\_SUPPORT\_OLD\_TARS",

FALSE), tar = Sys.getenv("TAR"))

## Inside utils::untar()

if (inherits(tarfile, "connection") || identical(tar, "internal")) {

if (!missing(compressed))

warning("argument 'compressed' is ignored for the internal method")

return(untar2(tarfile, files, list, exdir, restore\_times))

}

## Further below

TAR <- tar

if (!nzchar(TAR) && .Platform$OS.type == "windows" && nzchar(Sys.which("tar.exe")))

TAR <- "tar.exe"

if (!nzchar(TAR) || TAR == "internal")

return(untar2(tarfile, files, list, exdir))

In this case, the first untar2() call is called. That is: return(untar2(tarfile, files, list, exdir, restore\_times)). The error message incomplete block on file is not really informative in this case because untar2() is not happy when there’s a space in the path to the file [3](https://feeds.feedburner.com/2019/09/18/windows-user-space-issues-with-installing-r-packages/#fn3).

We can get around this untar2() issue by uncompressing the tar.gz file ourselves in a path that has no spaces. For example, if we download DO.db\_2.9.tar.gz to C:\R we can uncompress the tar.gz file with:

utils::untar('C:/R/DO.db\_2.9.tar.gz')

**Installation**

Let’s proceed to installing the package.

> install.packages('C:/R/DO.db', repos = NULL, type = 'source', lib = 'C:/R/R-3.6.0/library')

\* installing \*source\* package 'DO.db' ...

\*\* using staged installation

\*\* R

\*\* inst

\*\* byte-compile and prepare package for lazy loading

ARGUMENT 'Collado\AppData\Local\Temp\Rtmp8EQDjB\Rin2ef05088650f' \_\_ignored\_\_

Error: object 'ÿþ' not found

Execution halted

ERROR: lazy loading failed for package 'DO.db'

\* removing 'C:/R/R-3.6.0/library/DO.db'

Warning message:

In install.packages("C:/R/DO.db", repos = NULL, type = "source", :

installation of package ‘C:/R/DO.db’ had non-zero exit status

>

Oh noes! It didn’t work What happened?

If you look closely, you’ll see that it says ARGUMENT 'Collado\AppData\Local\Temp\Rtmp8EQDjB\Rin2ef05088650f' \_\_ignored\_\_. Wait! Collado is the Lastname portion of the username! So we have another space issue [4](https://feeds.feedburner.com/2019/09/18/windows-user-space-issues-with-installing-r-packages/#fn4). That structure though looks very familiar, it’s from base::tempdir()!

> tempdir()

[1] "C:\\Users\\Leo Collado\\AppData\\Local\\Temp\\RtmpqiBJ53"

The help file for ?tempdir contained the clues to solving this issue.

By default, tmpdir will be the directory given by tempdir().  
This will be a subdirectory of the per-session temporary directory  
found by the following rule when the R session is started.  
The environment variables TMPDIR, TMP and TEMP are checked in turn  
and the first found which points to a writable directory is used:  
if none succeeds ‘/tmp’ is used. The path should not contain spaces.  
Note that setting any of these environment variables in the R session  
has no effect on tempdir(): the per-session temporary directory is  
created before the interpreter is started.

We can set the TMPDIR environment variable which will be used by the R session spawned inside the installation of [*DO.db*](https://bioconductor.org/packages/3.9/DO.db) and… it works!

> Sys.setenv(TMPDIR = 'C:/R/tmp\_leo')

> Sys.getenv('TMPDIR')

[1] "C:/R/tmp\_leo"

>

>

> install.packages('C:/R/DO.db', repos = NULL, type = 'source', lib = 'C:/R/R-3.6.0/library')

\* installing \*source\* package 'DO.db' ...

\*\* using staged installation

\*\* R

\*\* inst

\*\* byte-compile and prepare package for lazy loading

\*\* help

\*\*\* installing help indices

converting help for package 'DO.db'

finding HTML links ... done

DOANCESTOR html

DOBASE html

DOCHILDREN html

DOMAPCOUNTS html

DOOBSOLETE html

DOOFFSPRING html

DOPARENTS html

DOSYNONYM html

DOTERM html

DOTerms-class html

DOTermsAnnDbBimap html

DO\_dbconn html

\*\* building package indices

\*\* testing if installed package can be loaded from temporary location

\*\*\* arch - i386

\*\*\* arch - x64

\*\* testing if installed package can be loaded from final location

\*\*\* arch - i386

\*\*\* arch - x64

\*\* testing if installed package keeps a record of temporary installation path

\* DONE (DO.db)

Making 'packages.html' ... done

**clusterProfiler installation**

Now we can continue and install [*clusterProfiler*](https://bioconductor.org/packages/3.9/clusterProfiler), right?

> BiocManager::install('clusterProfiler', lib = 'C:/R/R-3.6.0/library')

Bioconductor version 3.9 (BiocManager 1.30.4), R 3.6.0 (2019-04-26)

Installing package(s) 'clusterProfiler'

also installing the dependencies ‘sys’, ‘formatR’, ‘askpass’, ‘farver’, ‘backports’, ‘zeallot’, ‘lambda.r’, ‘futile.options’, ‘curl’, ‘mime’, ‘openssl’, ‘hms’, ‘triebeard’, ‘tweenr’, ‘polyclip’, ‘RcppEigen’, ‘colorspace’, ‘utf8’, ‘vctrs’, ‘futile.logger’, ‘snow’, ‘data.table’, ‘fastmatch’, ‘stringr’, ‘httr’, ‘jsonlite’, ‘progress’, ‘urltools’, ‘xml2’, ‘gridGraphics’, ‘ggforce’, ‘ggrepel’, ‘viridis’, ‘labeling’, ‘munsell’, ‘R6’, ‘cli’, ‘crayon’, ‘fansi’, ‘pillar’, ‘BiocParallel’, ‘fgsea’, ‘reshape2’, ‘cowplot’, ‘europepmc’, ‘ggplotify’, ‘ggraph’, ‘ggridges’, ‘gridExtra’, ‘igraph’, ‘purrr’, ‘RColorBrewer’, ‘UpSetR’, ‘gtable’, ‘lazyeval’, ‘rlang’, ‘scales’, ‘tibble’, ‘viridisLite’, ‘withr’, ‘dplyr’, ‘glue’, ‘stringi’, ‘tidyselect’, ‘DOSE’, ‘enrichplot’, ‘ggplot2’, ‘GO.db’, ‘GOSemSim’, ‘plyr’, ‘qvalue’, ‘rvcheck’, ‘tidyr’

## Delete more output

The downloaded binary packages are in

C:\Users\Leo Collado\AppData\Local\Temp\RtmpqiBJ53\downloaded\_packages

installing the source packages ‘pillar’, ‘GO.db’

trying URL '<https://cloud.r-project.org/src/contrib/pillar_1.4.1.tar.gz>'

Content type 'application/x-gzip' length 228572 bytes (223 KB)

downloaded 223 KB

trying URL '<https://bioconductor.org/packages/3.9/data/annotation/src/contrib/GO.db_3.8.2.tar.gz>'

Content type 'application/x-gzip' length 31820866 bytes (30.3 MB)

downloaded 30.3 MB

Error in untar2(tarfile, files, list, exdir, restore\_times) :

incomplete block on file

Error in untar2(tarfile, files, list, exdir, restore\_times) :

incomplete block on file

The downloaded source packages are in

‘C:\Users\Leo Collado\AppData\Local\Temp\RtmpqiBJ53\downloaded\_packages’

The issue is again that utils:::untar2() and thus utils::untar() does not like spaces in the paths. If we look at where the packages were downloaded more closely, we can see a space there at C:\Users\Leo Collado\AppData\Local\Temp\RtmpqiBJ53\downloaded\_packages. If you check the help file for utils::install.packages() you’ll see that destdir controls this:

destdir

directory where downloaded packages are stored. If it is NULL  
(the default) a subdirectory downloaded\_packages of the session  
temporary directory will be used (and the files will be deleted  
at the end of the session).

If we dig into utils::install.packages() we can see how this comes to play.

## Part of utils::install.packages()

if (is.null(destdir) && nonlocalrepos) {

tmpd <- file.path(tempdir(), "downloaded\_packages")

if (!file.exists(tmpd) && !dir.create(tmpd))

stop(gettextf("unable to create temporary directory %s",

sQuote(tmpd)), domain = NA)

}

> BiocManager::install('clusterProfiler', lib = 'C:/R/R-3.6.0/library', destdir = 'C:/R/dest\_leo')

Bioconductor version 3.9 (BiocManager 1.30.4), R 3.6.0 (2019-04-26)

Installing package(s) 'clusterProfiler'

also installing the dependency ‘GO.db’

trying URL '<https://bioconductor.org/packages/3.9/bioc/bin/windows/contrib/3.6/clusterProfiler_3.12.0.zip>'

Content type 'application/zip' length 623524 bytes (608 KB)

downloaded 608 KB

package ‘clusterProfiler’ successfully unpacked and MD5 sums checked

installing the source package ‘GO.db’

trying URL '<https://bioconductor.org/packages/3.9/data/annotation/src/contrib/GO.db_3.8.2.tar.gz>'

Content type 'application/x-gzip' length 31820866 bytes (30.3 MB)

downloaded 30.3 MB

\* installing \*source\* package 'GO.db' ...

\*\* using staged installation

\*\* R

\*\* inst

\*\* byte-compile and prepare package for lazy loading

\*\* help

\*\*\* installing help indices

converting help for package 'GO.db'

finding HTML links ... done

GOBASE html

GOBPANCESTOR html

GOBPCHILDREN html

GOBPOFFSPRING html

GOBPPARENTS html

GOCCANCESTOR html

GOCCCHILDREN html

GOCCOFFSPRING html

GOCCPARENTS html

GOMAPCOUNTS html

GOMFANCESTOR html

GOMFCHILDREN html

GOMFOFFSPRING html

GOMFPARENTS html

GOOBSOLETE html

GOSYNONYM html

GOTERM html

GO\_dbconn html

\*\* building package indices

\*\* testing if installed package can be loaded from temporary location

\*\*\* arch - i386

\*\*\* arch - x64

\*\* testing if installed package can be loaded from final location

\*\*\* arch - i386

\*\*\* arch - x64

\*\* testing if installed package keeps a record of temporary installation path

\* DONE (GO.db)

Making 'packages.html' ... done

**Closing**

All of the above seemed like too much. In addition, it seemed like BiocManager::install('hypeR', destdir = 'C:/R/dest\_leo') was not working [6](https://feeds.feedburner.com/2019/09/18/windows-user-space-issues-with-installing-r-packages/#fn6). I likely missed something here earlier today. So controlling utils::tempdir() seemed like the easiest solution such that the defaults of where a package gets downloaded, uncompressed, etc all worked. And the simplest solution we thought of was to create the C:\TEMP directory and update the Windows environment variables TMP and TEMP to [point](https://www.howtogeek.com/285710/how-to-move-windows-temporary-folders-to-another-drive/) to that location. Then, the rest of the commands worked without having to specify lib or destdir or manually run utils::untar().

As a whole, remember to look for spaces in the error messages! This is specially relevant when you are having issues as a Microsoft Windows R user.

If you have other solutions for Microsoft Windows R users with usernames that have at least one space, please let us know in the comments! Thank you! 