

Krill's loader quick setup guide for Windows users

This guide shall provide the necessary information about getting Krill's loader compiled on a Windows system using Cygwin.

Chapter one – installing Cygwin

-download Cygwin from <https://www.cygwin.com/>

Current Cygwin DLL version

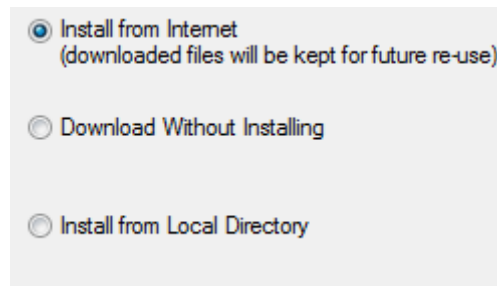
The most recent version of the Cygwin DLL is [2.6.1](#). Install it by running [setup-x86.exe](#) (32-bit installation) or [setup-x86_64.exe](#) (64-bit installation).

Use the setup program to perform a [fresh install](#) or to [update](#) an existing installation.

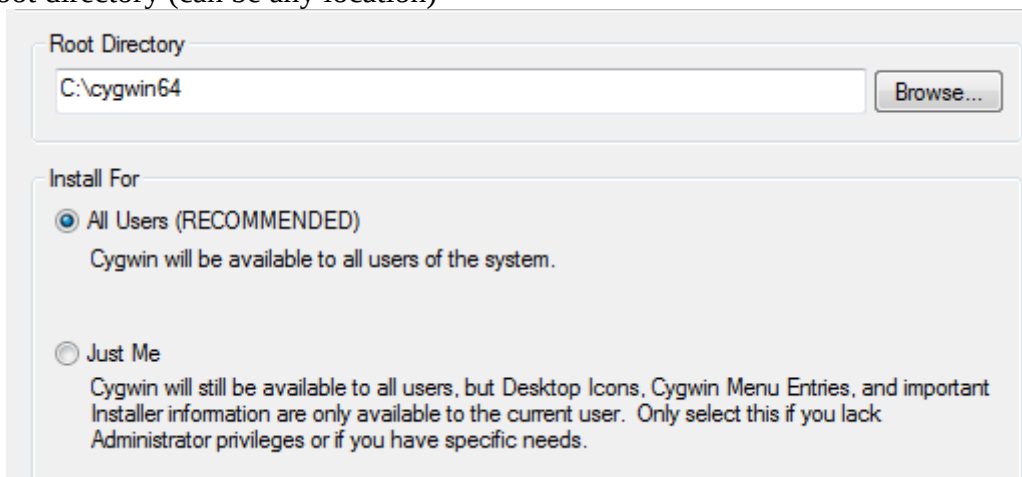
Since in most cases the 64-bit version will be the interesting one this will be used in the examples.

- make sure to have a good internet connection and run 'setup-x86_64.exe'

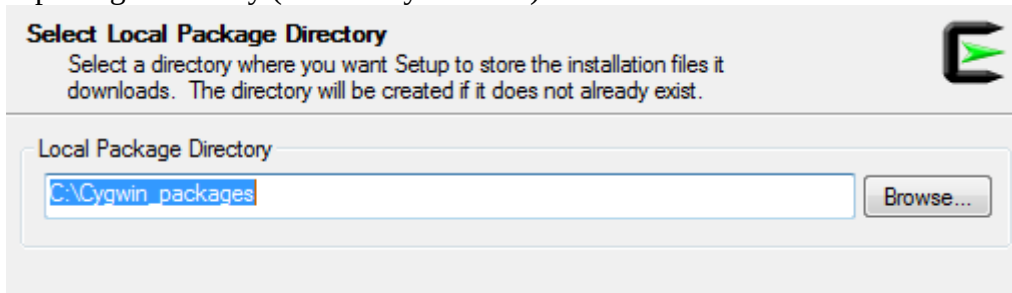
-select 'Install from internet'



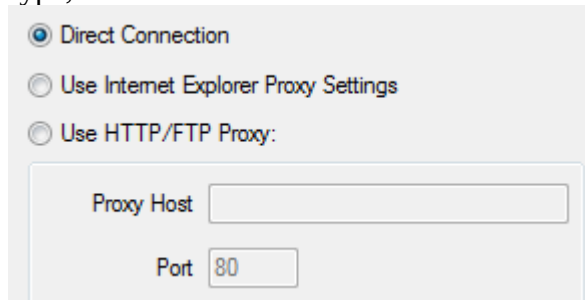
-choose root directory (can be any location)



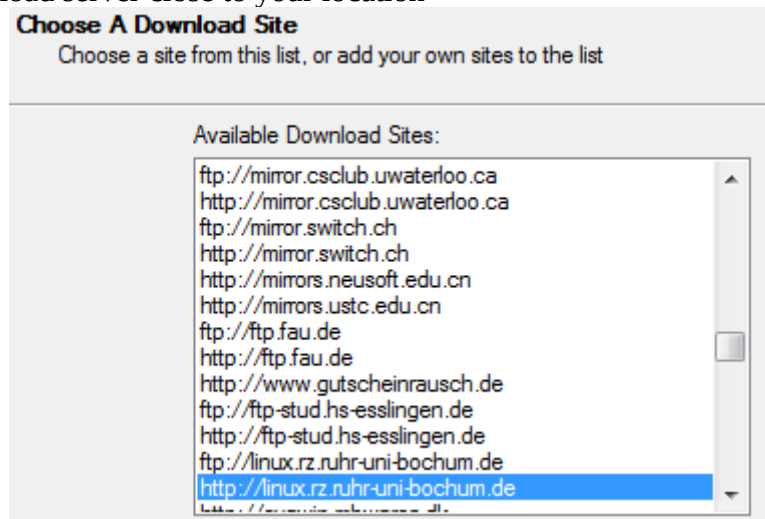
-select local package directory (can be any location)



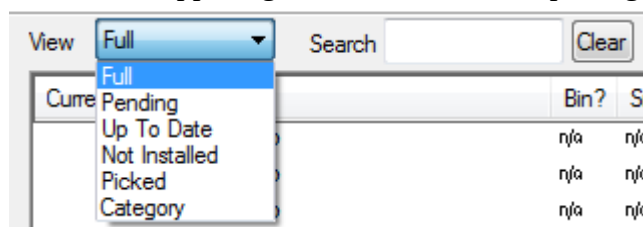
-next select the connection type, in most cases this is 'direct connection'



-then select a download server close to your location

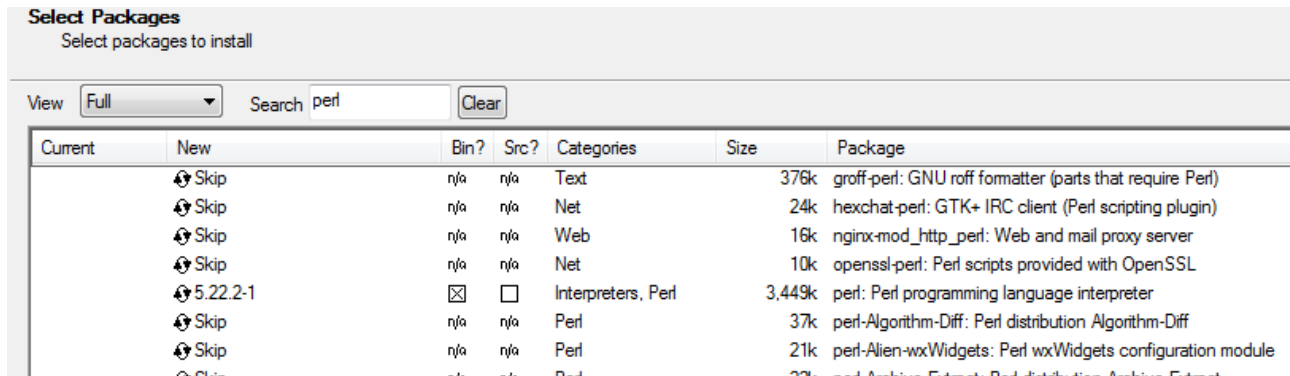


-on the next screen select 'full' in the upper right corner to make all packages visible in the list

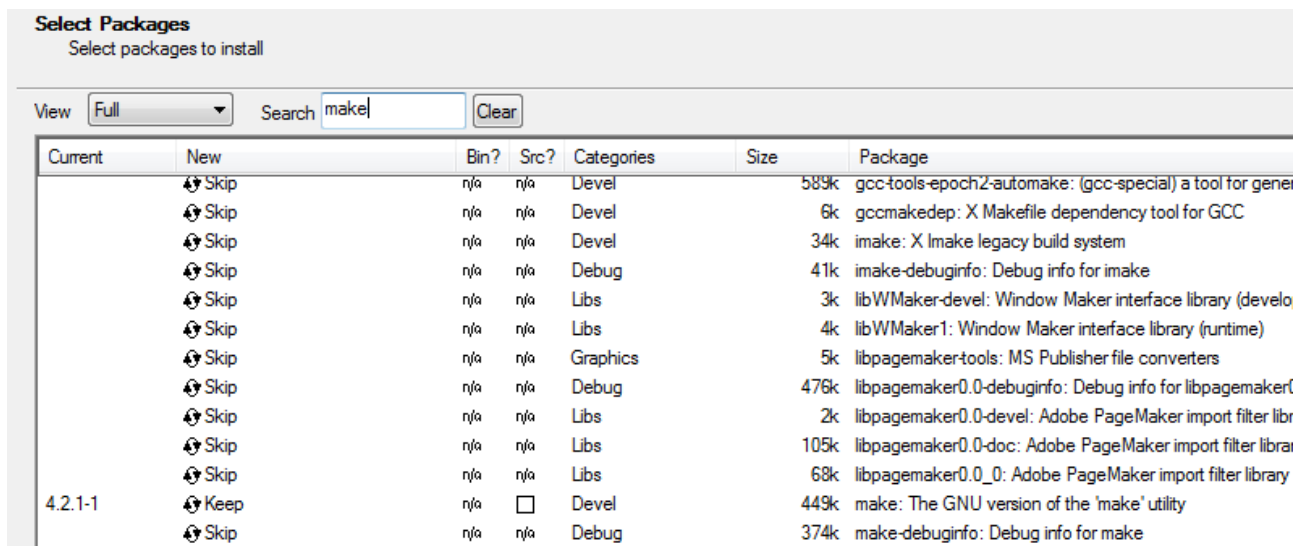


-two additional packages are needed for building the Krill's loader, namely 'perl' and 'make'

-> type 'perl' in the search box and click on the word 'skip' in the line for the perl interpreter until it shows the version number



->repeat this with 'make'. The Picture below shows how the package selection screen looks if the package 'make' was previously already installed. In that situation just keep 'keep'. Otherwise do like above.



-Installation can take quite some minutes. In my case it took even several hours since there seems to be a strange bug where a process dash.exe needs extremely long to complete. Usually this should not be the case.

- during installation cygwin will create a home directory for the current user which can be found for the user 'map' (in my case) at this location:

C:\cygwin64\home\map

In cygwin everything done by the user happens here by default.

- I recommend to uncomment some lines in the file .bashrc like this:

```
# Some shortcuts for different directory listings
alias ls='ls -hF --color=tty'           # classify files in colour
alias dir='ls --color=auto --format=vertical'
# alias vdir='ls --color=auto --format=long'
alias ll='ls -l'                         # long list
# alias la='ls -A'                       # all but . and ..
# alias l='ls -CF'                      #
```

Chapter two – installation of CC65

-navigate with your web-browser to: <https://cc65.github.io/cc65/>

It is possible to get and build the most recent version from a git repository but there is a very convenient short-cut available by just downloading the current snapshot, provided as a .zip file containing all the executables for windows.

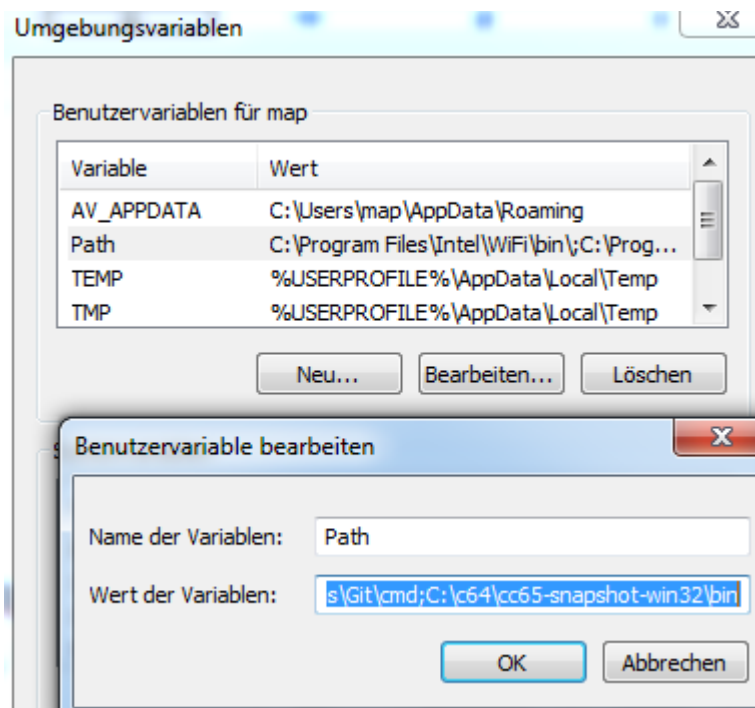
So scroll down and find the link in the Link section:

Links

- ▶ [Source Code](#)
- ▶ [Windows Snapshot](#)
- ▶ [Documentation](#)
- ▶ [Wiki](#)

-unpack the .zip to a directory of your choice:
eg. C:\c64\cc65-snapshot-win32

-then you need to add the subdirectory '\bin' (e.g. C:\c64\cc65-snapshot-win32\bin) of this directory to the 'path'



Chapter three – getting the Loader

- download Krill's loader zip package from <http://csdb.dk/release/?id=118713> and unzip into the home directory in cygwin root folder (example: C:\cygwin64\home\map)

- due to an incompatibility with the newer/current versions of cc65 a patch is needed in the file (can be skipped with loader revision #156 and newer):

C:\cygwin64\home\map\loader\src\make-loadersymbolsinc.pl
Many thanks goes to DJ Gruby for providing this information.

1. Open file "/loader/src/make-loadersymbolsinc.pl" for editing.

2. Change line 21 from:

```
if ($i =~ /list:/) {
```

to:

```
if ($i =~ /(?:Exports|Segment) list/) {
```

3. Change line 43 from:

```
if ($current_list =~ 'Exports list:') {
```

to:

```
if ($current_list =~ 'Exports list by (?:name|value):') {
```

4. Change line 51 from:

```
} elsif ($i =~ /(\w+)\s+(\w+)\s+(\w+)/) {
```

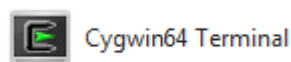
to:

```
} elsif ($i =~ /(\w+)\s+(\w+)\s+(\w+[\^:]*$)/) {
```

- although there is much more mentioned in the prerequisites.txt, the loader can now already be compiled

Chapter four – building the loader

-start the cygwin terminal



-navigate to /loader/src and execute

the following line: make prg INSTALL=2000 RESIDENT=0800 ZP=10

```
~/loader/src  
map@Marvin3 ~  
$ cd loader  
map@Marvin3 ~/loader  
$ cd src  
map@Marvin3 ~/loader/src  
$ ll  
insgesamt 28K  
drwxr-xr-x+ 1 map None    0 29. Jan 20:56 decompress/  
drwxr-xr-x+ 1 map None    0 29. Jan 20:56 drives/  
drwxr-xr-x+ 1 map None    0 29. Jan 20:56 hal/  
drwxr-xr-x+ 1 map None    0 29. Jan 20:56 install/  
-r-xr-xr-x  1 map None  13K 28. Apr 2013 Makefile*  
-rwxr-xr-x  1 map None  2,0K  2. Feb 03:51 make-loadersymbolsinc.pl*  
drwxr-xr-x+ 1 map None    0 29. Jan 20:56 resident/  
map@Marvin3 ~/loader/src  
$ make prg INSTALL=2000 RESIDENT=0800 ZP=10
```

-Finally the result of the successful build can be found in C:\cygwin64\home\map\loader\build

Basically only three files are of interest

install-c64.prg (in this example now @\$2000)

loader-c64.prg (in this example now @\$0800)

loadersymbols-c64.inc (for information only, where to find the entry calls for the init and loader as well as the used ZP addresses)

Chapter five - Configuration

The standard configuration might likely not fit in every case.

All configuration options can be found in the file:

C:\cygwin64\home\map\loader\include\config.inc

Modify as needed and run the make in /src again as explained above.

The loader gives a lot of options and features, therefore it would be a long task to describe a recipe for each possibility but as a first guidance I would **highly recommend** to follow the tutorials from DJ Gruby in Attitude #15, which can be downloaded from here: <http://csdb.dk/release/?id=133118>

These tutorials provide as well a very good example how to integrate and use the loader in the own production, which is a very good starting point.