New chat

Today

You said:

I want to include: D:/Programs/msys64/ucrt64/share to the search path in msys. Can I do it in .bash\_profile or .bashrc. My .bash\_profile:

# To the extent possible under law, the author(s) have dedicated all

# copyright and related and neighboring rights to this software to the

# public domain worldwide. This software is distributed without any warranty.

# You should have received a copy of the CC0 Public Domain Dedication along

# with this software.

# If not, see <https://creativecommons.org/publicdomain/zero/1.0/>.

# ~/.bash\_profile: executed by bash(1) for login shells.

# The copy in your home directory (~/.bash\_profile) is yours, please

# feel free to customise it to create a shell

# environment to your liking. If you feel a change

# would be beneficial to all, please feel free to send

# a patch to the msys2 mailing list.

# User dependent .bash\_profile file

# source the users bashrc if it exists

if [ -f "${HOME}/.bashrc" ] ; then

source "${HOME}/.bashrc"

fi

# Set PATH so it includes user's private bin if it exists

# if [ -d "${HOME}/bin" ] ; then

# PATH="${HOME}/bin:${PATH}"

# fi

# Set MANPATH so it includes users' private man if it exists

# if [ -d "${HOME}/man" ]; then

# MANPATH="${HOME}/man:${MANPATH}"

# fi

# Set INFOPATH so it includes users' private info if it exists

# if [ -d "${HOME}/info" ]; then

# INFOPATH="${HOME}/info:${INFOPATH}"

# fi

Copilot said:

Copilot

Yes, you can add the directory to the search path in either .bash\_profile or .bashrc. Since .bash\_profile is executed for login shells and .bashrc for non-login shells, it's common practice to add such configurations to .bashrc and source it from .bash\_profile.

Here's how you can modify your .bash\_profile and .bashrc:

Modify .bashrc: Add the following line to your .bashrc file:

export PATH="$PATH:/d/Programs/msys64/ucrt64/share"

Modify .bash\_profile: Ensure your .bash\_profile sources .bashrc (which it already does in your case):

if [ -f "${HOME}/.bashrc" ] ; then

source "${HOME}/.bashrc"

fi

This way, the path will be included regardless of whether you start a login or non-login shell.

Would you like any further assistance with this?

Edit in Pages

AI-generated content may be incorrect

You said:

This is my .bashrc:

# To the extent possible under law, the author(s) have dedicated all

# copyright and related and neighboring rights to this software to the

# public domain worldwide. This software is distributed without any warranty.

# You should have received a copy of the CC0 Public Domain Dedication along

# with this software.

# If not, see <https://creativecommons.org/publicdomain/zero/1.0/>.

# ~/.bashrc: executed by bash(1) for interactive shells.

# The copy in your home directory (~/.bashrc) is yours, please

# feel free to customise it to create a shell

# environment to your liking. If you feel a change

# would be benifitial to all, please feel free to send

# a patch to the msys2 mailing list.

# User dependent .bashrc file

# If not running interactively, don't do anything

[[ "$-" != \*i\* ]] && return

# Shell Options

#

# See man bash for more options...

#

# Don't wait for job termination notification

# set -o notify

#

# Don't use ^D to exit

# set -o ignoreeof

#

# Use case-insensitive filename globbing

# shopt -s nocaseglob

#

# Make bash append rather than overwrite the history on disk

# shopt -s histappend

#

# When changing directory small typos can be ignored by bash

# for example, cd /vr/lgo/apaache would find /var/log/apache

# shopt -s cdspell

# Completion options

#

# These completion tuning parameters change the default behavior of bash\_completion:

#

# Define to access remotely checked-out files over passwordless ssh for CVS

# COMP\_CVS\_REMOTE=1

#

# Define to avoid stripping description in --option=description of './configure --help'

# COMP\_CONFIGURE\_HINTS=1

#

# Define to avoid flattening internal contents of tar files

# COMP\_TAR\_INTERNAL\_PATHS=1

#

# Uncomment to turn on programmable completion enhancements.

# Any completions you add in ~/.bash\_completion are sourced last.

# [[ -f /etc/bash\_completion ]] && . /etc/bash\_completion

# History Options

#

# Don't put duplicate lines in the history.

# export HISTCONTROL=$HISTCONTROL${HISTCONTROL+:}ignoredups

#

# Ignore some controlling instructions

# HISTIGNORE is a colon-delimited list of patterns which should be excluded.

# The '&' is a special pattern which suppresses duplicate entries.

# export HISTIGNORE=$'[ \t]\*:&:[fb]g:exit'

# export HISTIGNORE=$'[ \t]\*:&:[fb]g:exit:ls' # Ignore the ls command as well

#

# Whenever displaying the prompt, write the previous line to disk

# export PROMPT\_COMMAND="history -a"

# Aliases

#

# Some people use a different file for aliases

# if [ -f "${HOME}/.bash\_aliases" ]; then

# source "${HOME}/.bash\_aliases"

# fi

#

# Some example alias instructions

# If these are enabled they will be used instead of any instructions

# they may mask. For example, alias rm='rm -i' will mask the rm

# application. To override the alias instruction use a \ before, ie

# \rm will call the real rm not the alias.

#

# Interactive operation...

# alias rm='rm -i'

# alias cp='cp -i'

# alias mv='mv -i'

#

# Default to human readable figures

# alias df='df -h'

# alias du='du -h'

#

# Misc :)

# alias less='less -r' # raw control characters

# alias whence='type -a' # where, of a sort

# alias grep='grep --color' # show differences in colour

# alias egrep='egrep --color=auto' # show differences in colour

# alias fgrep='fgrep --color=auto' # show differences in colour

#

# 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' #

# Umask

#

# /etc/profile sets 022, removing write perms to group + others.

# Set a more restrictive umask: i.e. no exec perms for others:

# umask 027

# Paranoid: neither group nor others have any perms:

# umask 077

# Functions

#

# Some people use a different file for functions

# if [ -f "${HOME}/.bash\_functions" ]; then

# source "${HOME}/.bash\_functions"

# fi

#

# Some example functions:

#

# a) function settitle

# settitle ()

# {

# echo -ne "\e]2;$@\a\e]1;$@\a";

# }

#

# b) function cd\_func

# This function defines a 'cd' replacement function capable of keeping,

# displaying and accessing history of visited directories, up to 10 entries.

# To use it, uncomment it, source this file and try 'cd --'.

# acd\_func 1.0.5, 10-nov-2004

# Petar Marinov, http:/geocities.com/h2428, this is public domain

# cd\_func ()

# {

# local x2 the\_new\_dir adir index

# local -i cnt

#

# if [[ $1 == "--" ]]; then

# dirs -v

# return 0

# fi

#

# the\_new\_dir=$1

# [[ -z $1 ]] && the\_new\_dir=$HOME

#

# if [[ ${the\_new\_dir:0:1} == '-' ]]; then

# #

# # Extract dir N from dirs

# index=${the\_new\_dir:1}

# [[ -z $index ]] && index=1

# adir=$(dirs +$index)

# [[ -z $adir ]] && return 1

# the\_new\_dir=$adir

# fi

#

# #

# # '~' has to be substituted by ${HOME}

# [[ ${the\_new\_dir:0:1} == '~' ]] && the\_new\_dir="${HOME}${the\_new\_dir:1}"

#

# #

# # Now change to the new dir and add to the top of the stack

# pushd "${the\_new\_dir}" > /dev/null

# [[ $? -ne 0 ]] && return 1

# the\_new\_dir=$(pwd)

#

# #

# # Trim down everything beyond 11th entry

# popd -n +11 2>/dev/null 1>/dev/null

#

# #

# # Remove any other occurence of this dir, skipping the top of the stack

# for ((cnt=1; cnt <= 10; cnt++)); do

# x2=$(dirs +${cnt} 2>/dev/null)

# [[ $? -ne 0 ]] && return 0

# [[ ${x2:0:1} == '~' ]] && x2="${HOME}${x2:1}"

# if [[ "${x2}" == "${the\_new\_dir}" ]]; then

# popd -n +$cnt 2>/dev/null 1>/dev/null

# cnt=cnt-1

# fi

# done

#

# return 0

# }

#

# alias cd=cd\_func

Copilot said:

Copilot

Great, thanks for sharing your .bashrc file! To add the directory to your search path, you can include the following line at the end of your .bashrc file:

export PATH="$PATH:/d/Programs/msys64/ucrt64/share"

This will ensure that the directory is added to your PATH every time you start an interactive shell.

After adding the line, you can either restart your terminal or source the .bashrc file to apply the changes immediately:

source ~/.bashrc

Let me know if you need any more help!

Edit in Pages

AI-generated content may be incorrect

You said:

Actually the real problem is different. I install gimp-RC3 in msys using pacman. But at the end of installation I got this message:

Note that 'D:/Programs/msys64/ucrt64/share' is not in the search path

set by the XDG\_DATA\_HOME and XDG\_DATA\_DIRS

environment variables, so applications may not

be able to find it until you set them. The

directories currently searched are:

- D:\Programs\msys64\home\Administrator\.local\share

- /usr/local/share/

- /usr/share/

Copilot said:

Copilot

To resolve this issue, you need to set the XDG\_DATA\_HOME and XDG\_DATA\_DIRS environment variables to include D:/Programs/msys64/ucrt64/share. You can do this by adding the following lines to your .bashrc file:

export XDG\_DATA\_HOME="$HOME/.local/share"

export XDG\_DATA\_DIRS="/usr/local/share:/usr/share:/d/Programs/msys64/ucrt64/share"

This will ensure that the directory is included in the search path for data files. After adding these lines, you can either restart your terminal or source the .bashrc file to apply the changes immediately:

source ~/.bashrc

This should help applications find the necessary files in D:/Programs/msys64/ucrt64/share.

Let me know if you need any more assistance!

Edit in Pages

AI-generated content may be incorrect

You said:

Still cannot load gimp:

# source ~/.bashrc

(venv313)

Administrator@WIN-GJS3SIOM8MB MSYS ~

# gimp

bash: gimp: command not found

(venv313)

Administrator@WIN-GJS3SIOM8MB MSYS ~

# gimp-3

bash: gimp-3: command not found

(venv313)

Copilot said:

Copilot

It looks like the GIMP executable isn't in your PATH. Let's try a few steps to troubleshoot this:

Verify Installation: Ensure that GIMP is installed correctly. You can check the installation directory to see if the GIMP executable is present.

Locate GIMP Executable: Use the find command to locate the GIMP executable:

find / -name gimp

Add GIMP to PATH: Once you locate the GIMP executable, add its directory to your PATH. For example, if the executable is in /usr/bin, you can add the following line to your .bashrc:

export PATH="$PATH:/usr/bin"

Check Environment Variables: Ensure that the XDG\_DATA\_HOME and XDG\_DATA\_DIRS variables are set correctly. You can verify this by running:

echo $XDG\_DATA\_HOME

echo $XDG\_DATA\_DIRS

Restart Terminal: After making these changes, restart your terminal or source the .bashrc file again:

source ~/.bashrc

If GIMP is still not found, it might be helpful to reinstall it using pacman:

pacman -S mingw-w64-x86\_64-gimp

Let me know if this helps or if you need further assistance!

Edit in Pages

AI-generated content may be incorrect

You said:

See the output of find command:

# find / -name gimp

/tmp/gimp

/ucrt64/etc/gimp

/ucrt64/lib/gimp

/ucrt64/share/gimp

/ucrt64/share/licenses/gimp

Issue is, gimp won't load only with executable because it will also need to search for libraries

Copilot said:

Copilot

Thanks for sharing the output. It looks like the GIMP executable and its libraries are installed in the /ucrt64 directory. To ensure GIMP can find all necessary libraries, you need to add the relevant directories to your PATH and LD\_LIBRARY\_PATH.

Here's how you can update your .bashrc file:

Add GIMP directories to PATH:

export PATH="$PATH:/ucrt64/bin"

Add GIMP libraries to LD\_LIBRARY\_PATH:

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/ucrt64/lib"

Ensure XDG\_DATA\_DIRS includes GIMP's share directory:

export XDG\_DATA\_DIRS="$XDG\_DATA\_DIRS:/ucrt64/share"

After adding these lines to your .bashrc, source the file to apply the changes:

source ~/.bashrc

Now, try running GIMP again:

gimp

This should help GIMP locate its libraries and run correctly. Let me know if you encounter any further issues!

Edit in Pages

AI-generated content may be incorrect

You said:

gimp crahses in msys:

# gimp

set device 'System Aggregated Pointer' to mode: disabled

(gimp.exe:5652): GLib-GIO-ERROR \*\*: 23:54:23.043: No GSettings schemas are installed on the system

D:\Programs\msys64\ucrt64\lib\gimp\3.0\plug-ins\script-fu\script-fu.exe: fatal error: GIMP crashed

(script-fu.exe:8252): LibGimp-WARNING \*\*: 23:54:26.464: script-fu.exe: gimp\_flush(): error: Invalid argument

Copilot said:

Copilot

It looks like GIMP is crashing due to missing GSettings schemas and issues with the script-fu plugin. Here are some steps to resolve these issues:

Install GSettings Schemas: You need to compile and install the GSettings schemas. Run the following command to compile the schemas:

glib-compile-schemas /ucrt64/share/glib-2.0/schemas

Ensure GSettings Environment Variables: Make sure the XDG\_DATA\_DIRS environment variable includes the directory where the schemas are located. Add this to your .bashrc:

export XDG\_DATA\_DIRS="$XDG\_DATA\_DIRS:/ucrt64/share"

Check for Missing Libraries: Ensure that all necessary libraries are in the LD\_LIBRARY\_PATH. Add this to your .bashrc:

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/ucrt64/lib"

Reinstall GIMP: If the above steps don't resolve the issue, try reinstalling GIMP using pacman:

pacman -S mingw-w64-x86\_64-gimp

After making these changes, source your .bashrc to apply the updates:

source ~/.bashrc

Then, try running GIMP again:

gimp

These steps should help resolve the issues with missing GSettings schemas and the script-fu plugin crash. Let me know if you need any further assistance!

Edit in Pages

AI-generated content may be incorrect

You said:

Problem is related to script-fu.exe in msys:

# glib-compile-schemas /ucrt64/share/glib-2.0/schemas

(venv313)

Administrator@WIN-GJS3SIOM8MB MSYS ~

# gimp

set device 'System Aggregated Pointer' to mode: disabled

(gimp.exe:6772): GLib-GIO-ERROR \*\*: 23:55:33.242: No GSettings schemas are installed on the system

D:\Programs\msys64\ucrt64\lib\gimp\3.0\plug-ins\script-fu\script-fu.exe: fatal error: GIMP crashed

(script-fu.exe:2616): LibGimp-WARNING \*\*: 23:55:37.003: script-fu.exe: gimp\_flush(): error: Invalid argument

(venv313)

Copilot said:

Copilot

It seems like the issue with script-fu.exe crashing is related to missing GSettings schemas and potentially other dependencies. Here are some steps to try and resolve this:

Ensure GSettings Schemas are Compiled: You mentioned running glib-compile-schemas, but let's make sure the schemas are in the correct directory:

glib-compile-schemas /ucrt64/share/glib-2.0/schemas

Set Environment Variables: Ensure that the XDG\_DATA\_DIRS environment variable includes the directory where the schemas are located. Add this to your .bashrc:

export XDG\_DATA\_DIRS="/ucrt64/share:/usr/local/share:/usr/share"

Check for Missing Libraries: Ensure that all necessary libraries are in the LD\_LIBRARY\_PATH. Add this to your .bashrc:

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/ucrt64/lib"

Reinstall GIMP: If the above steps don't resolve the issue, try reinstalling GIMP using pacman:

pacman -S mingw-w64-x86\_64-gimp

Run GIMP with Debugging: Run GIMP with debugging options to get more detailed error messages:

gimp --verbose --console-messages

After making these changes, source your .bashrc to apply the updates:

source ~/.bashrc

Then, try running GIMP again:

gimp

These steps should help resolve the issues with missing GSettings schemas and the script-fu plugin crash. Let me know if you need any further assistance!