

NAME

`gnome-extensions` - Command line tool for managing GNOME extensions

SYNOPSIS

`gnome-extensions help` [*COMMAND*]

`gnome-extensions version`

`gnome-extensions enable` *UUID*

`gnome-extensions disable` *UUID*

`gnome-extensions reset` *UUID*

`gnome-extensions info` *UUID*

`gnome-extensions show` *UUID*

`gnome-extensions list` [*OPTION...*]

`gnome-extensions prefs` *UUID*

`gnome-extensions create` [*OPTION...*]

`gnome-extensions pack` [*OPTION...*]

`gnome-extensions install` [*OPTION...*] *PACK*

`gnome-extensions uninstall` *UUID*

DESCRIPTION

`gnome-extensions` is a utility that makes some common GNOME extensions operations available on the command line.

COMMON OPTIONS

All commands except for **help** and **version** handle the following options:

--quiet, -q

Do not print error messages

COMMANDS

help [*COMMAND*]

Displays a short synopsis of the available commands or provides detailed help on a specific command.

version

Prints the program version.

enable *UUID*

Enables the extension identified by *UUID*.

The command will not detect any errors from the extension itself, use the **info** command to confirm that the extension state is **ENABLED**.

If the extension is already enabled, the command will do nothing.

disable *UUID*

Disables the extension identified by *UUID*.

If the extension is not enabled, the command will do nothing.

reset *UUID*

Reset the extension identified by *UUID*.

The extension will be disabled in GNOME, but may be enabled by other sessions like GNOME Classic.

info *UUID*

Show details of the extension identified by *UUID*, including name, description and state.

show *UUID*

Synonym of **info**.

list [*OPTION...*]

Displays a list of installed extensions.

Options

--user

Include extensions installed in the user's **\$HOME**

--system

Include extensions installed in the system

--enabled

Include enabled extensions

--disabled

Include disabled extensions

--prefs

Only include extensions with preferences

--updates

Only include extensions with pending updates

-d, --details

Show some extra information for each extension

prefs *UUID*

Open the preference dialog of the extension identified by *UUID*.

create [*OPTION...*]

Creates a new extension from a template.

Options**--name=***NAME*

Set the user-visible name in the extension's metadata to *NAME*

--description=*DESC*

Set the description in the extension's metadata to *DESC*

--uuid=*UUID*

Set the unique extension ID in the metadata to *UUID*

--template=*TEMPLATE*

Use *TEMPLATE* as base for the new extension

-i, --interactive

Prompt for any extension metadata that hasn't been provided on the command line

pack [*OPTION...*] [*SOURCE-DIRECTORY*]

Creates an extension bundle that is suitable for publishing.

The bundle will always include the required files `extension.js` and `metadata.json`, as well as the optional `stylesheet.css` and `prefs.js` if found. Each additional source that should be included must be specified with **--extra-source**.

If the extension includes one or more GSettings schemas, they can either be placed in a `schemas/` folder to be picked up automatically, or be specified with **--schema**.

Similarly, translations are included automatically when they are located in a `po/` folder, otherwise the **--podir** option can be used to point to the correct directory. If no gettext domain is provided on the command line, the value of the **gettext-domain** metadata field is used if it exists, and the extension UUID if not.

All files are searched in *SOURCE-DIRECTORY* if specified, or the current directory otherwise.

Options

--extra-source=FILE

Additional source to include in the bundle

--schema=SCHEMA

A GSettings schema that should be compiled and included

--podir=PODIR

A directory with translations that should be compiled and included

--gettext-domain=DOMAIN

The gettext domain to use for translations

-f, --force

Overwrite an existing pack

-o, --out-dir=DIRECTORY

The directory where the pack should be created

install [*OPTION...*] *PACK*

Installs an extension from the bundle *PACK*.

The command unpacks the extension files and moves them to the expected location in the user's

\$HOME, so that it will be loaded in the next session.

It is mainly intended for testing, not as a replacement for the extension website. As extensions have privileged access to the user's session, it is advised to never load extensions from untrusted sources without carefully reviewing their content.

Options

--force

Override an existing extension

uninstall *UUID*

Uninstalls the extension identified by *UUID*.

EXIT STATUS

On success 0 is returned, a non-zero failure code otherwise.

BUGS

The tool is part of the gnome-shell project, and bugs should be reported in its issue tracker at <https://gitlab.gnome.org/GNOME/gnome-shell/issues>.