

[Skip to content](#)

Preferences Operators

bpy.ops.preferences.addon_disable(*, module="")

Turn off this add-on

PARAMETERS:

module (*string, (optional, never None)*) – Module, Module name of the add-on to disable

FILE:

[startup/bl_operators/userpref.py:529](#)

bpy.ops.preferences.addon_enable(*, module="")

Turn on this add-on

PARAMETERS:

module (*string, (optional, never None)*) – Module, Module name of the add-on to enable

FILE:

[startup/bl_operators/userpref.py:465](#)

bpy.ops.preferences.addon_expand(*, module="")

Display information and preferences for this add-on

PARAMETERS:

module (*string, (optional, never None)*) – Module, Module name of the add-on to expand

FILE:

[startup/bl_operators/userpref.py:907](#)

bpy.ops.preferences.addon_install(*, overwrite=True, enable_on_install=False, target="", filepath="", filter_folder=True, filter_python=True, filter_glob='*.py;*.zip')

Install an add-on

PARAMETERS:

- **overwrite** (*boolean, (optional)*) – Overwrite, Remove existing add-ons with the same ID
- **enable_on_install** (*boolean, (optional)*) – Enable on Install, Enable after installing
- **target** (*enum in [], (optional)*) – Target Path
- **filepath** (*string, (optional, never None)*) – filepath
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_python** (*boolean, (optional)*) – Filter Python
- **filter_glob** (*string, (optional, never None)*) – filter_glob

FILE:

[startup/bl_operators/userpref.py:695](#)

bpy.ops.preferences.addon_refresh()

Scan add-on directories for new modules

FILE:

[startup/bl_operators/userpref.py:628](#)

bpy.ops.preferences.addon_remove(*, module="")

Delete the add-on from the file system

PARAMETERS:

module (*string, (optional, never None)*) – Module, Module name of the add-on to remove

FILE:

[startup/bl_operators/userpref.py:856](#)

```
bpy.ops.preferences.addon_show(*, module='')
```

Show add-on preferences

PARAMETERS:

module (*string, (optional, never None)*) – Module, Module name of the add-on to expand

FILE:

[startup/bl_operators/userpref.py:933](#)

```
bpy.ops.preferences.app_template_install(*, overwrite=True, filepath='', filter_folder=True, filter_glob='*.zip')
```

Install an application template

PARAMETERS:

- **overwrite** (*boolean, (optional)*) – Overwrite, Remove existing template with the same ID
- **filepath** (*string, (optional, never None)*) – filepath
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_glob** (*string, (optional, never None)*) – filter_glob

FILE:

[startup/bl_operators/userpref.py:980](#)

```
bpy.ops.preferences.asset_library_add(*, directory='', hide_props_region=True, check_existing=False, filter_blender=False,
filter_backup=False, filter_image=False, filter_movie=False, filter_python=False, filter_font=False, filter_sound=False,
filter_text=False, filter_archive=False, filter_btx=False, filter_collada=False, filter_alembic=False, filter_usd=False,
filter_obj=False, filter_volume=False, filter_folder=True, filter_blenlib=False, filemode=9, display_type='DEFAULT',
sort_method='')
```

Add a directory to be used by the Asset Browser as source of assets

PARAMETERS:

- **directory** (*string, (optional, never None)*) – Directory, Directory of the file
- **hide_props_region** (*boolean, (optional)*) – Hide Operator Properties, Collapse the region displaying the operator settings
- **check_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter_blender** (*boolean, (optional)*) – Filter .blend files
- **filter_backup** (*boolean, (optional)*) – Filter .blend files
- **filter_image** (*boolean, (optional)*) – Filter image files
- **filter_movie** (*boolean, (optional)*) – Filter movie files
- **filter_python** (*boolean, (optional)*) – Filter Python files
- **filter_font** (*boolean, (optional)*) – Filter font files
- **filter_sound** (*boolean, (optional)*) – Filter sound files
- **filter_text** (*boolean, (optional)*) – Filter text files
- **filter_archive** (*boolean, (optional)*) – Filter archive files
- **filter_btx** (*boolean, (optional)*) – Filter btx files
- **filter_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter_alembic** (*boolean, (optional)*) – Filter Alembic files
- **filter_usd** (*boolean, (optional)*) – Filter USD files
- **filter_obj** (*boolean, (optional)*) – Filter OBJ files
- **filter_volume** (*boolean, (optional)*) – Filter OpenVDB volume files
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_blenlib** (*boolean, (optional)*) – Filter Blender IDs
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file

- **display_type** (*enum in ['DEFAULT', 'LIST_VERTICAL', 'LIST_HORIZONTAL', 'THUMBNAIL'], (optional)*) – Display Type
 - `DEFAULT` Default – Automatically determine display type for files.
 - `LIST_VERTICAL` Short List – Display files as short list.
 - `LIST_HORIZONTAL` Long List – Display files as a detailed list.
 - `THUMBNAIL` Thumbnails – Display files as thumbnails.
- **sort_method** (*enum in [], (optional)*) – File sorting mode

`bpy.ops.preferences.asset_library_remove(*, index=0)`

Remove a path to a .blend file, so the Asset Browser will not attempt to show it anymore

PARAMETERS:

index (*int in [0, inf], (optional)*) – Index

`bpy.ops.preferences.associate_blend()`

Use this installation for .blend files and to display thumbnails

`bpy.ops.preferences.autoexec_path_add()`

Add path to exclude from auto-execution

`bpy.ops.preferences.autoexec_path_remove(*, index=0)`

Remove path to exclude from auto-execution

PARAMETERS:

index (*int in [0, inf], (optional)*) – Index

`bpy.ops.preferences.copy_prev()`

Copy settings from previous version

FILE:

[startup/bl_operators/userpref.py:170](#)

`bpy.ops.preferences.extension_repo_add(*, name="", remote_url="", use_access_token=False, access_token="", use_sync_on_startup=False, use_custom_directory=False, custom_directory="", type='REMOTE')`

Add a new repository used to store extensions

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Unique repository name
- **remote_url** (*string, (optional, never None)*) – URL, Remote URL to the extension repository, the file-system may be referenced using the file URI scheme: “file://”
- **use_access_token** (*boolean, (optional)*) – Requires Access Token, Repository requires an access token
- **access_token** (*string, (optional, never None)*) – Secret, Personal access token, may be required by some repositories
- **use_sync_on_startup** (*boolean, (optional)*) – Check for Updates on Startup, Allow Blender to check for updates upon launch
- **use_custom_directory** (*boolean, (optional)*) – Custom Directory, Manually set the path for extensions to be stored. When disabled a user’s extensions directory is created.
- **custom_directory** (*string, (optional, never None)*) – Custom Directory, The local directory containing extensions
- **type** (*enum in ['REMOTE', 'LOCAL'], (optional)*) – Type, The kind of repository to add
 - `REMOTE` Add Remote Repository – Add a repository referencing a remote repository with support for listing and updating extensions.
 - `LOCAL` Add Local Repository – Add a repository managed manually without referencing an external repository.

`bpy.ops.preferences.extension_repo_remove(*, index=0, remove_files=False)`

Remove an extension repository

PARAMETERS:

- **index** (*int in [0, inf], (optional)*) – Index
- **remove_files** (*boolean, (optional)*) – Remove Files, Remove extension files when removing the repository

bpy.ops.preferences.extension_url_drop(*, url="")

Handle dropping an extension URL

PARAMETERS:

url (*string, (optional, never None)*) – URL, Location of the extension to install

bpy.ops.preferences.keyconfig_activate(*, filepath="")

Undocumented, consider [contributing](#).

PARAMETERS:

filepath (*string, (optional, never None)*) – filepath

FILE:

[startup/bl_operators/userpref.py:91](#)

bpy.ops.preferences.keyconfig_export(*, all=False, filepath="", filter_folder=True, filter_text=True, filter_python=True)

Export key configuration to a Python script

PARAMETERS:

- **all** (*boolean, (optional)*) – All Keymaps, Write all keymaps (not just user modified)
- **filepath** (*string, (optional, never None)*) – filepath
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_text** (*boolean, (optional)*) – Filter text
- **filter_python** (*boolean, (optional)*) – Filter Python

FILE:

[startup/bl_operators/userpref.py:307](#)

bpy.ops.preferences.keyconfig_import(*, filepath='keymap.py', filter_folder=True, filter_text=True, filter_python=True, keep_original=True)

Import key configuration from a Python script

PARAMETERS:

- **filepath** (*string, (optional, never None)*) – filepath
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_text** (*boolean, (optional)*) – Filter text
- **filter_python** (*boolean, (optional)*) – Filter Python
- **keep_original** (*boolean, (optional)*) – Keep Original, Keep original file after copying to configuration folder

FILE:

[startup/bl_operators/userpref.py:236](#)

bpy.ops.preferences.keyconfig_remove()

Remove key config

FILE:

[startup/bl_operators/userpref.py:445](#)

bpy.ops.preferences.keyconfig_test()

Test key configuration for conflicts

FILE:

[startup/bl_operators/userpref.py:194](#)

[startup/bl_operators/userpref.py:177](#)

bpy.ops.preferences.**keyitem_add()**

Add key map item

FILE:

[startup/bl_operators/userpref.py:393](#)

bpy.ops.preferences.**keyitem_remove(*, item_id=0)**

Remove key map item

PARAMETERS:

item_id (*int in [-inf, inf], (optional)*) – Item Identifier, Identifier of the item to remove

FILE:

[startup/bl_operators/userpref.py:425](#)

bpy.ops.preferences.**keyitem_restore(*, item_id=0)**

Restore key map item

PARAMETERS:

item_id (*int in [-inf, inf], (optional)*) – Item Identifier, Identifier of the item to restore

FILE:

[startup/bl_operators/userpref.py:378](#)

bpy.ops.preferences.**keymap_restore(*, all=False)**

Restore key map(s)

PARAMETERS:

all (*boolean, (optional)*) – All Keymaps, Restore all keymaps to default

FILE:

[startup/bl_operators/userpref.py:349](#)

bpy.ops.preferences.**reset_default_theme()**

Reset to the default theme colors

bpy.ops.preferences.**script_directory_add(*, directory="", filter_folder=True)**

Undocumented, consider [contributing](#).

PARAMETERS:

- **directory** (*string, (optional, never None)*) – directory
- **filter_folder** (*boolean, (optional)*) – Filter Folders

FILE:

[startup/bl_operators/userpref.py:1235](#)

bpy.ops.preferences.**script_directory_remove(*, index=0)**

Undocumented, consider [contributing](#).

PARAMETERS:

index (*int in [-inf, inf], (optional)*) – Index, Index of the script directory to remove

FILE:

[startup/bl_operators/userpref.py:1265](#)

bpy.ops.preferences.**studiolight_copy_settings(*, index=0)**

Copy Studio Light settings to the Studio Light editor

PARAMETERS:

index (*int in [-inf, inf], (optional)*) – index

FILE:

[startup/bl_operators/userpref.py:1206](#)

```
bpy.ops.preferences.studiolight_install(*, files=None, directory="", filter_folder=True, filter_glob='*.png;*.jpg;*.hdr;*.exr',  
type='MATCAP')
```

Install a user defined light

PARAMETERS:

- **files** (*bpy_prop_collection of OperatorFileListElement, (optional)*) – File Path
- **directory** (*string, (optional, never None)*) – directory
- **filter_folder** (*boolean, (optional)*) – Filter Folders
- **filter_glob** (*string, (optional, never None)*) – filter_glob
- **type** (*enum in ['MATCAP', 'WORLD', 'STUDIO'], (optional)*) – Type
 - **MATCAP** MatCap – Install custom MatCaps.
 - **WORLD** World – Install custom HDRIs.
 - **STUDIO** Studio – Install custom Studio Lights.

FILE:

[startup/bl_operators/userpref.py:1090](#)

```
bpy.ops.preferences.studiolight_new(*, filename='StudioLight')
```

Save custom studio light from the studio light editor settings

PARAMETERS:

filename (*string, (optional, never None)*) – Name

FILE:

[startup/bl_operators/userpref.py:1136](#)

```
bpy.ops.preferences.studiolight_uninstall(*, index=0)
```

Delete Studio Light

PARAMETERS:

index (*int in [-inf, inf], (optional)*) – index

FILE:

[startup/bl_operators/userpref.py:1187](#)

```
bpy.ops.preferences.theme_install(*, overwrite=True, filepath="", filter_folder=True, filter_glob='*.xml')
```

Load and apply a Blender XML theme file

PARAMETERS:

- **overwrite** (*boolean, (optional)*) – Overwrite, Remove existing theme file if exists
- **filepath** (*string, (optional, never None)*) – filepath
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_glob** (*string, (optional, never None)*) – filter_glob

FILE:

[startup/bl_operators/userpref.py:581](#)

```
bpy.ops.preferences.unassociate_blend()
```

Remove this installation's associations with .blend files

