Skip to content

Asset Operators

bpy.ops.asset.**assign_action()**

Set this pose Action as active Action on the active Object

FILE:

addons_core/pose_library/operators.py:103

bpy.ops.asset.**bundle_install(*, asset_library_reference='', filepath='', hide_props_region=True, check_existing=True, filter_blender=True 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=8, display_type='DEFAULT', sort_method='')**

Copy the current .blend file into an Asset Library. Only works on standalone .blend files (i.e. when no other files are referenced)

PARAMETERS:

- **asset_library_reference** (*enum in [], (optional)*) – asset_library_reference
- **filepath** (*string, (optional, never None)*) – File Path, Path to 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.asset.**catalog_delete(*, catalog_id='')**

Remove an asset catalog from the asset library (contained assets will not be affected and show up as unassigned)

PARAMETERS:

catalog_id (*string, (optional, never None)*) – Catalog ID, ID of the catalog to delete

bpy.ops.asset.**catalog_new(*, parent_path='')**

Create a new catalog to put assets in

PARAMETERS:

parent_path (*string, (optional, never None)*) – Parent Path, Optional path defining the location to put the new catalog under

bpy.ops.asset.**catalog_redo()**

Redo the last undone edit to the asset catalogs

bpy.ops.asset.**catalog_undo()**

Undo the last edit to the asset catalogs

bpy.ops.asset.**catalog_undo_push()**

Store the current state of the asset catalogs in the undo buffer

bpy.ops.asset.**catalogs_save()**

Make any edits to any catalogs permanent by writing the current set up to the asset library

bpy.ops.asset.**clear(*, set_fake_user=False)**

Delete all asset metadata and turn the selected asset data-blocks back into normal data-blocks

PARAMETERS:

set_fake_user (*boolean, (optional)*) – Set Fake User, Ensure the data-block is saved, even when it is no longer marked as asset

bpy.ops.asset.**clear_single(*, set_fake_user=False)**

Delete all asset metadata and turn the asset data-block back into a normal data-block

PARAMETERS:

set_fake_user (*boolean, (optional)*) – Set Fake User, Ensure the data-block is saved, even when it is no longer marked as asset

bpy.ops.asset.**library_refresh()**

Reread assets and asset catalogs from the asset library on disk

bpy.ops.asset.**mark()**

Enable easier reuse of selected data-blocks through the Asset Browser, with the help of customizable metadata (like previews, descriptions and tag

bpy.ops.asset.**mark_single()**

Enable easier reuse of a data-block through the Asset Browser, with the help of customizable metadata (like previews, descriptions and tags)

bpy.ops.asset.**open_containing_blend_file()**

Open the blend file that contains the active asset

FILE:

startup/bl_operators/assets.py:103

bpy.ops.asset.**tag_add()**

Add a new keyword tag to the active asset

FILE:

startup/bl_operators/assets.py:42

bpy.ops.asset.**tag_remove()**

Remove an existing keyword tag from the active asset

FILE:

FILE:

startup/bl_operators/assets.py:65

Previous
Armature Operators

Ne
Boid Operat

Report issue on this page