

[Skip to content](#)

# Anim Operators

bpy.ops.anim.change\_frame(\*, frame=0.0, snap=False)

Interactively change the current frame number

## PARAMETERS:

- **frame** (*float in  $[-1.04857e+06, 1.04857e+06]$ , (optional)*) – Frame
- **snap** (*boolean, (optional)*) – Snap

bpy.ops.anim.channel\_select\_keys(\*, extend=False)

Select all keyframes of channel under mouse

## PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend, Extend selection

bpy.ops.anim.channel\_view\_pick(\*, include\_handles=True, use\_preview\_range=True)

Reset viewable area to show the channel under the cursor

## PARAMETERS:

- **include\_handles** (*boolean, (optional)*) – Include Handles, Include handles of keyframes when calculating extents
- **use\_preview\_range** (*boolean, (optional)*) – Use Preview Range, Ignore frames outside of the preview range

bpy.ops.anim.channels\_bake(\*, range=(0, 0), step=1.0, remove\_outside\_range=False, interpolation\_type='BEZIER', bake\_modifiers=True)

Create keyframes following the current shape of F-Curves of selected channels

## PARAMETERS:

- **range** (*int array of 2 items in  $[-inf, inf]$ , (optional)*) – Frame Range, The range in which to create new keys
- **step** (*float in  $[0.01, inf]$ , (optional)*) – Frame Step, At which interval to add keys
- **remove\_outside\_range** (*boolean, (optional)*) – Remove Outside Range, Removes keys outside the given range, leaving only the newly baked
- **interpolation\_type** (*enum in  $['BEZIER', 'LIN', 'CONST']$ , (optional)*) – Interpolation Type, Choose the interpolation type with which new keys will be added
  - **BEZIER** Bézier – New keys will be Bézier.
  - **LIN** Linear – New keys will be linear.
  - **CONST** Constant – New keys will be constant.
- **bake\_modifiers** (*boolean, (optional)*) – Bake Modifiers, Bake Modifiers into keyframes and delete them after

bpy.ops.anim.channels\_clean\_empty()

Delete all empty animation data containers from visible data-blocks

bpy.ops.anim.channels\_click(\*, extend=False, extend\_range=False, children\_only=False)

Handle mouse clicks over animation channels

## PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend Select
- **extend\_range** (*boolean, (optional)*) – Extend Range, Selection of active channel to clicked channel
- **children\_only** (*boolean, (optional)*) – Select Children Only

bpy.ops.anim.channels\_collapse(\*, all=True)

Collapse (close) all selected expandable animation channels

## PARAMETERS:

#### PARAMETERS:

**all** (*boolean, (optional)*) – All, Collapse all channels (not just selected ones)

bpy.ops.anim.channels\_delete()

Delete all selected animation channels

bpy.ops.anim.channels\_editable\_toggle(\*, mode='TOGGLE', type='PROTECT')

Toggle editability of selected channels

#### PARAMETERS:

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode
- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

bpy.ops.anim.channels\_expand(\*, all=True)

Expand (open) all selected expandable animation channels

#### PARAMETERS:

**all** (*boolean, (optional)*) – All, Expand all channels (not just selected ones)

bpy.ops.anim.channels\_fcurves\_enable()

Clear ‘disabled’ tag from all F-Curves to get broken F-Curves working again

bpy.ops.anim.channels\_group(\*, name='New Group')

Add selected F-Curves to a new group

#### PARAMETERS:

**name** (*string, (optional, never None)*) – Name, Name of newly created group

bpy.ops.anim.channels\_move(\*, direction='DOWN')

Rearrange selected animation channels

#### PARAMETERS:

**direction** (*enum in ['TOP', 'UP', 'DOWN', 'BOTTOM'], (optional)*) – Direction

bpy.ops.anim.channels\_rename()

Rename animation channel under mouse

bpy.ops.anim.channels\_select\_all(\*, action='TOGGLE')

Toggle selection of all animation channels

#### PARAMETERS:

**action** (*enum in ['TOGGLE', 'SELECT', 'DESELECT', 'INVERT'], (optional)*) –

Action, Selection action to execute

- **TOGGLE** Toggle – Toggle selection for all elements.
- **SELECT** Select – Select all elements.
- **DESELECT** Deselect – Deselect all elements.
- **INVERT** Invert – Invert selection of all elements.

bpy.ops.anim.channels\_select\_box(\*, xmin=0, xmax=0, ymin=0, ymax=0, wait\_for\_input=True, deselect=False, extend=True)

Select all animation channels within the specified region

#### PARAMETERS:

- **xmin** (*int in [-inf, inf], (optional)*) – X Min
- **xmax** (*int in [-inf, inf], (optional)*) – X Max
- **ymin** (*int in [-inf, inf], (optional)*) – Y Min

- **y<sub>max</sub>** (*int in [-inf, inf], (optional)*) –  $y_{\max}$
- **wait\_for\_input** (*boolean, (optional)*) – Wait for Input
- **deselect** (*boolean, (optional)*) – Deselect, Deselect rather than select items
- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first

**bpy.ops.anim.channels\_select\_filter()**

Start entering text which filters the set of channels shown to only include those with matching names

**bpy.ops.anim.channels\_setting\_disable(\*, mode='DISABLE', type='PROTECT')**

Disable specified setting on all selected animation channels

**PARAMETERS:**

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode
- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

**bpy.ops.anim.channels\_setting\_enable(\*, mode='ENABLE', type='PROTECT')**

Enable specified setting on all selected animation channels

**PARAMETERS:**

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode
- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

**bpy.ops.anim.channels\_setting\_toggle(\*, mode='TOGGLE', type='PROTECT')**

Toggle specified setting on all selected animation channels

**PARAMETERS:**

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode
- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

**bpy.ops.anim.channels\_ungroup()**

Remove selected F-Curves from their current groups

**bpy.ops.anim.channels\_view\_selected(\*, include\_handles=True, use\_preview\_range=True)**

Reset viewable area to show the selected channels

**PARAMETERS:**

- **include\_handles** (*boolean, (optional)*) – Include Handles, Include handles of keyframes when calculating extents
- **use\_preview\_range** (*boolean, (optional)*) – Use Preview Range, Ignore frames outside of the preview range

**bpy.ops.anim.clear\_useless\_actions(\*, only\_unused=True)**

Mark actions with no F-Curves for deletion after save and reload of file preserving “action libraries”

**PARAMETERS:**

- **only\_unused** (*boolean, (optional)*) – Only Unused, Only unused (Fake User only) actions get considered

**FILE:**

[startup/bl\\_operators/anim.py:351](#)

**bpy.ops.anim.convert\_legacy\_action()**

Convert a legacy Action to a layered Action on the active object

**bpy.ops.anim.copy\_driver\_button()**

Copy the driver for the highlighted button

**bpy.ops.anim.driver\_button\_add()**

Add driver for the property under the cursor

bpy.ops.anim.driver\_button\_edit()

Edit the drivers for the connected property represented by the highlighted button

bpy.ops.anim.driver\_button\_remove(\*, all=True)

Remove the driver(s) for the connected property(s) represented by the highlighted button

**PARAMETERS:**

**all** (*boolean, (optional)*) – All, Delete drivers for all elements of the array

bpy.ops.anim.end\_frame\_set()

Set the current frame as the preview or scene end frame

bpy.ops.anim.keyframe\_clear\_button(\*, all=True)

Clear all keyframes on the currently active property

**PARAMETERS:**

**all** (*boolean, (optional)*) – All, Clear keyframes from all elements of the array

bpy.ops.anim.keyframe\_clear\_v3d(\*, confirm=True)

Remove all keyframe animation for selected objects

**PARAMETERS:**

**confirm** (*boolean, (optional)*) – Confirm, Prompt for confirmation

bpy.ops.anim.keyframe\_delete(\*, type='DEFAULT')

Delete keyframes on the current frame for all properties in the specified Keying Set

**PARAMETERS:**

**type** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use

bpy.ops.anim.keyframe\_delete\_button(\*, all=True)

Delete current keyframe of current UI-active property

**PARAMETERS:**

**all** (*boolean, (optional)*) – All, Delete keyframes from all elements of the array

bpy.ops.anim.keyframe\_delete\_by\_name(\*, type='')

Alternate access to 'Delete Keyframe' for keymaps to use

**PARAMETERS:**

**type** (*string, (optional, never None)*) – Keying Set, The Keying Set to use

bpy.ops.anim.keyframe\_delete\_v3d(\*, confirm=True)

Remove keyframes on current frame for selected objects and bones

**PARAMETERS:**

**confirm** (*boolean, (optional)*) – Confirm, Prompt for confirmation

bpy.ops.anim.keyframe\_insert(\*, type='DEFAULT')

Insert keyframes on the current frame using either the active keying set, or the user preferences if no keying set is active

**PARAMETERS:**

**type** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use

bpy.ops.anim.keyframe\_insert\_button(\*, all=True)

Insert a keyframe for current UI-active property

#### PARAMETERS:

**all** (*boolean, (optional)*) – All, Insert a keyframe for all element of the array

bpy.ops.anim.keyframe\_insert\_by\_name(\*, type="")

Alternate access to 'Insert Keyframe' for keymaps to use

#### PARAMETERS:

**type** (*string, (optional, never None)*) – Keying Set, The Keying Set to use

bpy.ops.anim.keyframe\_insert\_menu(\*, type='DEFAULT', always\_prompt=False)

Insert Keyframes for specified Keying Set, with menu of available Keying Sets if undefined

#### PARAMETERS:

- **type** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use
- **always\_prompt** (*boolean, (optional)*) – Always Show Menu

bpy.ops.anim.keying\_set\_active\_set(\*, type='DEFAULT')

Set a new active keying set

#### PARAMETERS:

**type** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use

bpy.ops.anim.keying\_set\_add()

Add a new (empty) keying set to the active Scene

bpy.ops.anim.keying\_set\_export(\*, filepath="", filter\_folder=True, filter\_text=True, filter\_python=True)

Export Keying Set to 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

#### FILE:

[startup/bl\\_operators/anim.py:46](#)

bpy.ops.anim.keying\_set\_path\_add()

Add empty path to active keying set

bpy.ops.anim.keying\_set\_path\_remove()

Remove active Path from active keying set

bpy.ops.anim.keying\_set\_remove()

Remove the active keying set

bpy.ops.anim.keyingset\_button\_add(\*, all=True)

Add current UI-active property to current keying set

#### PARAMETERS:

**all** (*boolean, (optional)*) – All, Add all elements of the array to a Keying Set

bpy.ops.anim.keyingset\_button\_remove()

Remove current UI-active property from current keying set

bpy.ops.anim.merge\_animation()

Merge the animation of the selected objects into the action of the active object. Actions are not deleted by this, but might end up with zero users

`bpy.ops.anim.paste_driver_button()`

Paste the driver in the internal clipboard to the highlighted button

`bpy.ops.anim.previewrange_clear()`

Clear preview range

`bpy.ops.anim.previewrange_set(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True)`

Interactively define frame range used for playback

**PARAMETERS:**

- **xmin** (*int in  $[-inf, inf]$ , (optional)*) – X Min
- **xmax** (*int in  $[-inf, inf]$ , (optional)*) – X Max
- **ymin** (*int in  $[-inf, inf]$ , (optional)*) – Y Min
- **ymax** (*int in  $[-inf, inf]$ , (optional)*) – Y Max
- **wait\_for\_input** (*boolean, (optional)*) – Wait for Input

`bpy.ops.anim.scene_range_frame()`

Reset the horizontal view to the current scene frame range, taking the preview range into account if it is active

`bpy.ops.anim.separate_slots()`

Move all slots of the action on the active object into newly created, separate actions. All users of those slots will be reassigned to the new actions. The current action won't be deleted but will be empty and might end up having zero users

`bpy.ops.anim.slot_channels_move_to_new_action()`

Move the selected slots into a newly created action

`bpy.ops.anim.slot_new_for_id()`

Create a new action slot for this data-block, to hold its animation

**FILE:**

[startup/bl\\_operators/anim.py:705](#)

`bpy.ops.anim.slot_unassign_from_constraint()`

Un-assign the action slot from this constraint

**FILE:**

[startup/bl\\_operators/anim.py:762](#)

`bpy.ops.anim.slot_unassign_from_id()`

Un-assign the action slot, effectively making this data-block non-animated

**FILE:**

[startup/bl\\_operators/anim.py:741](#)

`bpy.ops.anim.slot_unassign_from_nla_strip()`

Un-assign the action slot from this NLA strip, effectively making it non-animated

**FILE:**

[startup/bl\\_operators/anim.py:762](#)

`bpy.ops.anim.start_frame_set()`

Set the current frame as the preview or scene start frame

`bpy.ops.anim.update_animated_transform_constraints(*, use_convert_to_radians=True)`

Update animated transform constraints of selected objects to use radians for rotation

Update f-curves/drivers affecting Transform constraints (use it with files from 2.70 and earlier)

**PARAMETERS:**

**use\_convert\_to\_radians** (*boolean, (optional)*) – Convert to Radians, Convert f-curves/drivers affecting rotations to radians. Warning: Use this only once

**FILE:**

[startup/bl\\_operators/anim.py:386](#)

`bpy.ops.anim.view_curve_in_graph_editor(*, all=False, isolate=False)`

Frame the property under the cursor in the Graph Editor

**PARAMETERS:**

- **all** (*boolean, (optional)*) – Show All, Frame the whole array property instead of only the index under the cursor
- **isolate** (*boolean, (optional)*) – Isolate, Hides all F-Curves other than the ones being framed