

[Skip to content](#)

Poselib Operators

`bpy.ops.poselib.apply_pose_asset(*, blend_factor=1.0, flipped=False)`

Apply the given Pose Action to the rig

PARAMETERS:

- **blend_factor** (*float in $[-inf, inf]$, (optional)*) – Blend Factor, Amount that the pose is applied on top of the existing poses. A negative value will subtract the pose instead of adding it
- **flipped** (*boolean, (optional)*) – Apply Flipped, When enabled, applies the pose flipped over the X-axis

`bpy.ops.poselib.asset_delete()`

Delete the selected Pose Asset

`bpy.ops.poselib.asset_modify(*, mode='ADJUST')`

Update the selected pose asset in the asset library from the currently selected bones. The mode defines how the asset is updated

PARAMETERS:

mode (*enum in $['ADJUST', 'REPLACE', 'ADD', 'REMOVE']$, (optional)*) –

Overwrite Mode, Specify which parts of the pose asset are overwritten

- **ADJUST** Adjust – Update existing channels in the pose asset but don't remove or add any channels.
- **REPLACE** Replace with Selection – Completely replace all channels in the pose asset with the current selection.
- **ADD** Add Selected Bones – Add channels of the selection to the pose asset. Existing channels will be updated.
- **REMOVE** Remove Selected Bones – Remove channels of the selection from the pose asset.

`bpy.ops.poselib.blend_pose_asset(*, blend_factor=0.0, flipped=False, release_confirm=False)`

Blend the given Pose Action to the rig

PARAMETERS:

- **blend_factor** (*float in $[-inf, inf]$, (optional)*) – Blend Factor, Amount that the pose is applied on top of the existing poses. A negative value will subtract the pose instead of adding it
- **flipped** (*boolean, (optional)*) – Apply Flipped, When enabled, applies the pose flipped over the X-axis
- **release_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.poselib.convert_old_object_poselib()`

Create a pose asset for each pose marker in this legacy pose library data-block

FILE:

[addons_core/pose_library/operators.py:342](#)

`bpy.ops.poselib.convert_old_poselib()`

Create a pose asset for each pose marker in the current action

FILE:

[addons_core/pose_library/operators.py:308](#)

`bpy.ops.poselib.copy_as_asset()`

Create a new pose asset on the clipboard, to be pasted into an Asset Browser

FILE:

[addons_core/pose_library/operators.py:116](#)

`bpy.ops.poselib.create_pose_asset(*, pose_name="", asset_library_reference="", catalog_path="", activate_new_action=False)`

Create a new asset from the selected bones in the scene

PARAMETERS:

- **pose_name** (*string, (optional, never None)*) – Pose Name, Name for the new pose asset
- **asset_library_reference** (*enum in [], (optional)*) – Library, Asset library used to store the new pose
- **catalog_path** (*string, (optional, never None)*) – Catalog, Catalog to use for the new asset
- **activate_new_action** (*boolean, (optional)*) – Activate New Action, This property is deprecated and will be removed in the future

`bpy.ops.poselib.paste_asset()`

Paste the Asset that was previously copied using Copy As Asset

FILE:

[addons_core/pose_library/operators.py:190](#)

`bpy.ops.poselib.pose_asset_select_bones(*, select=True, flipped=False)`

Select those bones that are used in this pose

PARAMETERS:

- **select** (*boolean, (optional)*) – Select
- **flipped** (*boolean, (optional)*) – Flipped

FILE:

[addons_core/pose_library/operators.py:228](#)

`bpy.ops.poselib.restore_previous_action()`

Switch back to the previous Action, after creating a pose asset

FILE:

[addons_core/pose_library/operators.py:65](#)