

# Export Scene Operators

```
bpy.ops.export_scene.fbx(*, filepath="", check_existing=True, filter_glob='*.fbx', use_selection=False, use_visible=False,
    use_active_collection=False, collection="", global_scale=1.0, apply_unit_scale=True, apply_scale_options='FBX_SCALE_NONE',
    use_space_transform=True, bake_space_transform=False, object_types={'ARMATURE', 'CAMERA', 'EMPTY', 'LIGHT',
    'MESH', 'OTHER'}, use_mesh_modifiers=True, use_mesh_modifiers_render=True, mesh_smooth_type='OFF',
    colors_type='SRGB', prioritize_active_color=False, use_subsurf=False, use_mesh_edges=False, use_tspace=False,
    use_triangles=False, use_custom_props=False, add_leaf_bones=True, primary_bone_axis='Y', secondary_bone_axis='X',
    use_armature_deform_only=False, armature_nodetype='NULL', bake_anim=True, bake_anim_use_all_bones=True,
    bake_anim_use_nla_strips=True, bake_anim_use_all_actions=True, bake_anim_force_startend_keying=True, bake_anim_step=1.
    bake_anim_simplify_factor=1.0, path_mode='AUTO', embed_textures=False, batch_mode='OFF', use_batch_own_dir=True,
    use_metadata=True, axis_forward='-Z', axis_up='Y')
```

Write a FBX file

## PARAMETERS:

- **filepath** (*string, (optional, never None)*) – File Path, Filepath used for exporting the file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_glob** (*string, (optional, never None)*) – filter\_glob
- **use\_selection** (*boolean, (optional)*) – Selected Objects, Export selected and visible objects only
- **use\_visible** (*boolean, (optional)*) – Visible Objects, Export visible objects only
- **use\_active\_collection** (*boolean, (optional)*) – Active Collection, Export only objects from the active collection (and its children)
- **collection** (*string, (optional, never None)*) – Source Collection, Export only objects from this collection (and its children)
- **global\_scale** (*float in [0.001, 1000], (optional)*) – Scale, Scale all data (Some importers do not support scaled armatures!)
- **apply\_unit\_scale** (*boolean, (optional)*) – Apply Unit, Take into account current Blender units settings (if unset, raw Blender Units values are used as-is)
- **apply\_scale\_options** (*enum in ['FBX\_SCALE\_NONE', 'FBX\_SCALE\_UNITS', 'FBX\_SCALE\_CUSTOM', 'FBX\_SCALE\_ALL'], (optional)*) –  
Apply Scalings, How to apply custom and units scalings in generated FBX file (Blender uses FBX scale to detect units on import, but many other applications do not handle the same way)
  - **FBX\_SCALE\_NONE** All Local – Apply custom scaling and units scaling to each object transformation, FBX scale remains at 1.0.
  - **FBX\_SCALE\_UNITS** FBX Units Scale – Apply custom scaling to each object transformation, and units scaling to FBX scale.
  - **FBX\_SCALE\_CUSTOM** FBX Custom Scale – Apply custom scaling to FBX scale, and units scaling to each object transformation.
  - **FBX\_SCALE\_ALL** FBX All – Apply custom scaling and units scaling to FBX scale.
- **use\_space\_transform** (*boolean, (optional)*) – Use Space Transform, Apply global space transform to the object rotations. When disabled only the axis space is written to the file and all object transforms are left as-is
- **bake\_space\_transform** (*boolean, (optional)*) – Apply Transform, Bake space transform into object data, avoids getting unwanted rotation to objects when target space is not aligned with Blender's space (WARNING! experimental option, use at own risk, known to be broken with armatures/animations)
- **object\_types** (*enum set in {'EMPTY', 'CAMERA', 'LIGHT', 'ARMATURE', 'MESH', 'OTHER'}, (optional)*) –  
Object Types, Which kind of object to export
  - **EMPTY** Empty.
  - **CAMERA** Camera.
  - **LIGHT** Lamp.
  - **ARMATURE** Armature – WARNING: not supported in dupli/group instances.
  - **MESH** Mesh.
  - **OTHER** Other – Other geometry types, like curve, metaball, etc. (converted to meshes).
- **use\_mesh\_modifiers** (*boolean, (optional)*) – Apply Modifiers, Apply modifiers to mesh objects (except Armature ones) - WARNING: prevents exporting shape keys
- **use\_mesh\_modifiers\_render** (*boolean, (optional)*) – Use Modifiers Render Setting, Use render settings when applying modifiers to mesh

objects (DISABLED in Blender 2.8)

- **mesh\_smooth\_type** (enum in ['OFF', 'FACE', 'EDGE'], (optional)) – Smoothing, Export smoothing information (prefer 'Normals Only' option if your target importer understand split normals)
  - OFF Normals Only – Export only normals instead of writing edge or face smoothing data.
  - FACE Face – Write face smoothing.
  - EDGE Edge – Write edge smoothing.
- **colors\_type** (enum in ['NONE', 'SRGB', 'LINEAR'], (optional)) – Vertex Colors, Export vertex color attributes
  - NONE None – Do not export color attributes.
  - SRGB sRGB – Export colors in sRGB color space.
  - LINEAR Linear – Export colors in linear color space.
- **prioritize\_active\_color** (boolean, (optional)) – Prioritize Active Color, Make sure active color will be exported first. Could be important since some other software can discard other color attributes besides the first one
- **use\_subsurf** (boolean, (optional)) – Export Subdivision Surface, Export the last Catmull-Rom subdivision modifier as FBX subdivision (do not apply the modifier even if 'Apply Modifiers' is enabled)
- **use\_mesh\_edges** (boolean, (optional)) – Loose Edges, Export loose edges (as two-vertices polygons)
- **use\_tspace** (boolean, (optional)) – Tangent Space, Add binormal and tangent vectors, together with normal they form the tangent space (will only work correctly with tris/quads only meshes!)
- **use\_triangles** (boolean, (optional)) – Triangulate Faces, Convert all faces to triangles
- **use\_custom\_props** (boolean, (optional)) – Custom Properties, Export custom properties
- **add\_leaf\_bones** (boolean, (optional)) – Add Leaf Bones, Append a final bone to the end of each chain to specify last bone length (use this when you intend to edit the armature from exported data)
- **primary\_bone\_axis** (enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)) – Primary Bone Axis
- **secondary\_bone\_axis** (enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)) – Secondary Bone Axis
- **use\_armature\_deform\_only** (boolean, (optional)) – Only Deform Bones, Only write deforming bones (and non-deforming ones when they have deforming children)
- **armature\_nodetype** (enum in ['NULL', 'ROOT', 'LIMBNODE'], (optional)) – Armature FBXNode Type, FBX type of node (object) used to represent Blender's armatures (use the Null type unless you experience issues with the other app, as other choices may not import back perfectly into Blender...)
  - NULL Null – 'Null' FBX node, similar to Blender's Empty (default).
  - ROOT Root – 'Root' FBX node, supposed to be the root of chains of bones...
  - LIMBNODE LimbNode – 'LimbNode' FBX node, a regular joint between two bones...
- **bake\_anim** (boolean, (optional)) – Baked Animation, Export baked keyframe animation
- **bake\_anim\_use\_all\_bones** (boolean, (optional)) – Key All Bones, Force exporting at least one key of animation for all bones (needed with some target applications, like UE4)
- **bake\_anim\_use\_nla\_strips** (boolean, (optional)) – NLA Strips, Export each non-muted NLA strip as a separated FBX's AnimStack, if any, instead of global scene animation
- **bake\_anim\_use\_all\_actions** (boolean, (optional)) – All Actions, Export each action as a separated FBX's AnimStack, instead of global scene animation (note that animated objects will get all actions compatible with them, others will get no animation at all)
- **bake\_anim\_force\_startend\_keying** (boolean, (optional)) – Force Start/End Keying, Always add a keyframe at start and end of actions for animated channels
- **bake\_anim\_step** (float in [0.01, 100], (optional)) – Sampling Rate, How often to evaluate animated values (in frames)
- **bake\_anim\_simplify\_factor** (float in [0, 100], (optional)) – Simplify, How much to simplify baked values (0.0 to disable, the higher the more simplified)
- **path\_mode** (enum in ['AUTO', 'ABSOLUTE', 'RELATIVE', 'MATCH', 'STRIP', 'COPY'], (optional)) – Path Mode, Method used to reference paths
  - AUTO Auto – Use relative paths with subdirectories only.
  - ABSOLUTE Absolute – Always write absolute paths.

- **RELATIVE** Relative – Always write relative paths (where possible).
- **MATCH** Match – Match absolute/relative setting with input path.
- **STRIP** Strip Path – Filename only.
- **COPY** Copy – Copy the file to the destination path (or subdirectory).
- **embed\_textures** (*boolean, (optional)*) – Embed Textures, Embed textures in FBX binary file (only for “Copy” path mode!)
- **batch\_mode** (*enum in ['OFF', 'SCENE', 'COLLECTION', 'SCENE\_COLLECTION', 'ACTIVE\_SCENE\_COLLECTION'], (optional)*)  
Batch Mode
  - **OFF** Off – Active scene to file.
  - **SCENE** Scene – Each scene as a file.
  - **COLLECTION** Collection – Each collection (data-block ones) as a file, does not include content of children collections.
  - **SCENE\_COLLECTION** Scene Collections – Each collection (including master, non-data-block ones) of each scene as a file, including content from children collections.
  - **ACTIVE\_SCENE\_COLLECTION** Active Scene Collections – Each collection (including master, non-data-block one) of the active scene as a file, including content from children collections.
- **use\_batch\_own\_dir** (*boolean, (optional)*) – Batch Own Dir, Create a dir for each exported file
- **use\_metadata** (*boolean, (optional)*) – Use Metadata
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up

#### FILE:

[addons\\_core/io\\_scene\\_fbx/\\_init\\_.py:578](#)

```
bpy.ops.export_scene.gltf(*, filepath="", check_existing=True, export_import_convert_lighting_mode='SPEC', gltf_export_id="",
export_use_gltfpack=False, export_gltfpack_tc=True, export_gltfpack_tq=8, export_gltfpack_si=1.0, export_gltfpack_sa=False,
export_gltfpack_slb=False, export_gltfpack_vp=14, export_gltfpack_vt=12, export_gltfpack_vn=8, export_gltfpack_vc=8,
export_gltfpack_vpi='Integer', export_gltfpack_nsq=True, export_gltfpack_kn=False, export_format="", ui_tab='GENERAL',
export_copyright="", export_image_format='AUTO', export_image_add_webp=False, export_image_webp_fallback=False,
export_texture_dir="", export_jpeg_quality=75, export_image_quality=75, export_keep_originals=False, export_texcoords=True,
export_normals=True, export_gn_mesh=False, export_draco_mesh_compression_enable=False,
export_draco_mesh_compression_level=6, export_draco_position_quantization=14, export_draco_normal_quantization=10,
export_draco_texcoord_quantization=12, export_draco_color_quantization=10, export_draco_generic_quantization=12,
export_tangents=False, export_materials='EXPORT', export_unused_images=False, export_unused_textures=False,
export_vertex_color='MATERIAL', export_all_vertex_colors=True, export_active_vertex_color_when_no_material=True,
export_attributes=False, use_mesh_edges=False, use_mesh_vertices=False, export_cameras=False, use_selection=False,
use_visible=False, use_renderable=False, use_active_collection_with_nested=True, use_active_collection=False,
use_active_scene=False, collection="", at_collection_center=False, export_extras=False, export_yup=True, export_apply=False,
export_shared_accessors=False, export_animations=True, export_frame_range=False, export_frame_step=1,
export_force_sampling=True, export_sampling_interpolation_fallback='LINEAR', export_pointer_animation=False,
export_animation_mode='ACTIONS', export_nla_strips_merged_animation_name='Animation', export_def_bones=False,
export_hierarchy_flatten_bones=False, export_hierarchy_flatten_objs=False, export_armature_object_remove=False,
export_leaf_bone=False, export_optimize_animation_size=True, export_optimize_animation_keep_anim_armature=True,
export_optimize_animation_keep_anim_object=False, export_optimize_disable_viewport=False, export_negative_frame='SLIDE',
export_anim_slide_to_zero=False, export_bake_animation=False, export_merge_animation='ACTION',
export_anim_single_armature=True, export_reset_pose_bones=True, export_current_frame=False,
export_rest_position_armature=True, export_anim_scene_split_object=True, export_skins=True, export_influence_nb=4,
export_all_influences=False, export_morph=True, export_morph_normal=True, export_morph_tangent=False,
export_morph_animation=True, export_morph_reset_sk_data=True, export_lights=False, export_try_sparse_sk=True,
export_try_onit_sparse_sk=False, export_gpu_instances=False, export_action_filter=False,
export_convert_animation_pointer=False, export_nla_strips=True, export_original_specular=False, will_save_settings=False,
export_hierarchy_full_collections=False, export_extra_animations=False, export_loglevel=-1, filter_glob='*.glb')
```

Export scene as glTF 2.0 file

#### PARAMETERS:

- **filepath** (*string, (optional, never None)*) – File Path. Filepath used for exporting the file.

- **niepau** (*string, (optional, never None)*) – nie pau, niepau used for exporting the nie
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **export\_import\_convert\_lighting\_mode** (*enum in ['SPEC', 'COMPAT', 'RAW'], (optional)*) – Lighting Mode, Optional backwards compatibility for non-standard render engines. Applies to lights
  - **SPEC** Standard – Physically-based glTF lighting units (cd, lx, nt).
  - **COMPAT** Unitless – Non-physical, unitless lighting. Useful when exposure controls are not available.
  - **RAW** Raw (Deprecated) – Blender lighting strengths with no conversion.
- **gltf\_export\_id** (*string, (optional, never None)*) – Identifier, Identifier of caller (in case of add-on calling this exporter). Can be useful in case of Extension added by other add-ons
- **export\_use\_gltfpack** (*boolean, (optional)*) – Use Gltfpack, Use gltfpack to simplify the mesh and/or compress its textures
- **export\_gltfpack\_tc** (*boolean, (optional)*) – KTX2 Compression, Convert all textures to KTX2 with BasisU supercompression
- **export\_gltfpack\_tq** (*int in [1, 10], (optional)*) – Texture Encoding Quality, Texture encoding quality
- **export\_gltfpack\_si** (*float in [0, 1], (optional)*) – Mesh Simplification Ratio, Simplify meshes targeting triangle count ratio
- **export\_gltfpack\_sa** (*boolean, (optional)*) – Aggressive Mesh Simplification, Aggressively simplify to the target ratio disregarding quality
- **export\_gltfpack\_slb** (*boolean, (optional)*) – Lock Mesh Border Vertices, Lock border vertices during simplification to avoid gaps on connected meshes
- **export\_gltfpack\_vp** (*int in [1, 16], (optional)*) – Position Quantization, Use N-bit quantization for positions
- **export\_gltfpack\_vt** (*int in [1, 16], (optional)*) – Texture Coordinate Quantization, Use N-bit quantization for texture coordinates
- **export\_gltfpack\_vn** (*int in [1, 16], (optional)*) – Normal/Tangent Quantization, Use N-bit quantization for normals and tangents
- **export\_gltfpack\_vc** (*int in [1, 16], (optional)*) – Vertex Color Quantization, Use N-bit quantization for colors
- **export\_gltfpack\_vpi** (*enum in ['Integer', 'Normalized', 'Floating-point'], (optional)*) – Vertex Position Attributes, Type to use for vertex position attributes
  - **Integer** Integer – Use integer attributes for positions.
  - **Normalized** Normalized – Use normalized attributes for positions.
  - **Floating-point** Floating-point – Use floating-point attributes for positions.
- **export\_gltfpack\_noq** (*boolean, (optional)*) – Disable Quantization, Disable quantization; produces much larger glTF files with no extension
- **export\_gltfpack\_kn** (*boolean, (optional)*) – Keep Named Nodes, Restrict some optimization to keep named nodes and meshes attached to named nodes so that named nodes can be transformed externally
- **export\_format** (*enum in [], (optional)*) – Format, Output format. Binary is most efficient, but JSON may be easier to edit later
- **ui\_tab** (*enum in ['GENERAL', 'MESHES', 'OBJECTS', 'ANIMATION'], (optional)*) – ui\_tab, Export setting categories
  - **GENERAL** General – General settings.
  - **MESHES** Meshes – Mesh settings.
  - **OBJECTS** Objects – Object settings.
  - **ANIMATION** Animation – Animation settings.
- **export\_copyright** (*string, (optional, never None)*) – Copyright, Legal rights and conditions for the model
- **export\_image\_format** (*enum in ['AUTO', 'JPEG', 'WEBP', 'NONE'], (optional)*) – Images, Output format for images. PNG is lossless and generally preferred, but JPEG might be preferable for web applications due to the smaller file size. Alternatively they can be omitted if they are not needed
  - **AUTO** Automatic – Save PNGs as PNGs, JPEGs as JPEGs, WebPs as WebPs. For other formats, use PNG.
  - **JPEG** JPEG Format (.jpg) – Save images as JPEGs. (Images that need alpha are saved as PNGs though.) Be aware of a possible loss in quality.
  - **WEBP** WebP Format – Save images as WebPs as main image (no fallback).
  - **NONE** None – Don't export images.
- **export\_image\_add\_webp** (*boolean, (optional)*) – Create WebP, Creates WebP textures for every texture. For already WebP textures, nothing happens
- **export\_image\_webp\_fallback** (*boolean, (optional)*) – WebP Fallback, For all WebP textures, create a PNG fallback texture

- **export\_texture\_dir** (*string, (optional, never None)*) – Textures, Folder to place texture files in. Relative to the .glTF file
- **export\_jpeg\_quality** (*int in [0, 100], (optional)*) – JPEG Quality, Quality of JPEG export
- **export\_image\_quality** (*int in [0, 100], (optional)*) – Image Quality, Quality of image export
- **export\_keep\_originals** (*boolean, (optional)*) – Keep Original, Keep original textures files if possible. WARNING: if you use more than one texture, where pbr standard requires only one, only one texture will be used. This can lead to unexpected results
- **export\_texcoords** (*boolean, (optional)*) – UVs, Export UVs (texture coordinates) with meshes
- **export\_normals** (*boolean, (optional)*) – Normals, Export vertex normals with meshes
- **export\_gn\_mesh** (*boolean, (optional)*) – Geometry Nodes Instances (Experimental), Export Geometry nodes instance meshes
- **export\_draco\_mesh\_compression\_enable** (*boolean, (optional)*) – Draco Mesh Compression, Compress mesh using Draco
- **export\_draco\_mesh\_compression\_level** (*int in [0, 10], (optional)*) – Compression Level, Compression level (0 = most speed, 6 = most compression, higher values currently not supported)
- **export\_draco\_position\_quantization** (*int in [0, 30], (optional)*) – Position Quantization Bits, Quantization bits for position values (0 = no quantization)
- **export\_draco\_normal\_quantization** (*int in [0, 30], (optional)*) – Normal Quantization Bits, Quantization bits for normal values (0 = no quantization)
- **export\_draco\_texcoord\_quantization** (*int in [0, 30], (optional)*) – Texcoord Quantization Bits, Quantization bits for texture coordinate values (0 = no quantization)
- **export\_draco\_color\_quantization** (*int in [0, 30], (optional)*) – Color Quantization Bits, Quantization bits for color values (0 = no quantization)
- **export\_draco\_generic\_quantization** (*int in [0, 30], (optional)*) – Generic Quantization Bits, Quantization bits for generic values like weight or joints (0 = no quantization)
- **export\_tangents** (*boolean, (optional)*) – Tangents, Export vertex tangents with meshes
- **export\_materials** (*enum in ['EXPORT', 'PLACEHOLDER', 'NONE'], (optional)*) – Materials, Export materials
  - **EXPORT** Export – Export all materials used by included objects.
  - **PLACEHOLDER** Placeholder – Do not export materials, but write multiple primitive groups per mesh, keeping material slot information.
  - **NONE** No export – Do not export materials, and combine mesh primitive groups, losing material slot information.
- **export\_unused\_images** (*boolean, (optional)*) – Unused Images, Export images not assigned to any material
- **export\_unused\_textures** (*boolean, (optional)*) – Prepare Unused Textures, Export image texture nodes not assigned to any material. This feature is not standard and needs an external extension to be included in the glTF file
- **export\_vertex\_color** (*enum in ['MATERIAL', 'ACTIVE', 'NONE'], (optional)*) – Use Vertex Color, How to export vertex color
  - **MATERIAL** Material – Export vertex color when used by material.
  - **ACTIVE** Active – Export active vertex color.
  - **NONE** None – Do not export vertex color.
- **export\_all\_vertex\_colors** (*boolean, (optional)*) – Export All Vertex Colors, Export all vertex colors, even if not used by any material. If no Vertex Color is used in the mesh materials, a fake COLOR\_0 will be created, in order to keep material unchanged
- **export\_active\_vertex\_color\_when\_no\_material** (*boolean, (optional)*) – Export Active Vertex Color When No Material, When there is no material on object, export active vertex color
- **export\_attributes** (*boolean, (optional)*) – Attributes, Export Attributes (when starting with underscore)
- **use\_mesh\_edges** (*boolean, (optional)*) – Loose Edges, Export loose edges as lines, using the material from the first material slot
- **use\_mesh\_vertices** (*boolean, (optional)*) – Loose Points, Export loose points as glTF points, using the material from the first material slot
- **export\_cameras** (*boolean, (optional)*) – Cameras, Export cameras
- **use\_selection** (*boolean, (optional)*) – Selected Objects, Export selected objects only
- **use\_visible** (*boolean, (optional)*) – Visible Objects, Export visible objects only
- **use\_renderable** (*boolean, (optional)*) – Renderable Objects, Export renderable objects only
- **use\_active\_collection\_with\_nested** (*boolean, (optional)*) – Include Nested Collections, Include active collection and nested collections
- **use\_active\_collection** (*boolean, (optional)*) – Active Collection, Export objects in the active collection only
- **use\_active\_scene** (*boolean, (optional)*) – Active Scene, Export active scene only

- **collection** (*string, (optional, never None)*) – Source Collection, Export only objects from this collection (and its children)
- **at\_collection\_center** (*boolean, (optional)*) – Export at Collection Center, Export at Collection center of mass of root objects of the collection
- **export\_extras** (*boolean, (optional)*) – Custom Properties, Export custom properties as glTF extras
- **export\_yup** (*boolean, (optional)*) – +Y Up, Export using glTF convention, +Y up
- **export\_apply** (*boolean, (optional)*) – Apply Modifiers, Apply modifiers (excluding Armatures) to mesh objects - WARNING: prevents exporting shape keys
- **export\_shared\_accessors** (*boolean, (optional)*) – Shared Accessors, Export Primitives using shared accessors for attributes
- **export\_animations** (*boolean, (optional)*) – Animations, Exports active actions and NLA tracks as glTF animations
- **export\_frame\_range** (*boolean, (optional)*) – Limit to Playback Range, Clips animations to selected playback range
- **export\_frame\_step** (*int in [1, 120], (optional)*) – Sampling Rate, How often to evaluate animated values (in frames)
- **export\_force\_sampling** (*boolean, (optional)*) – Always Sample Animations, Apply sampling to all animations
- **export\_sampling\_interpolation\_fallback** (*enum in ['LINEAR', 'STEP'], (optional)*) – Sampling Interpolation Fallback, Interpolation fallback for sampled animations, when the property is not keyed
  - **LINEAR** Linear – Linear interpolation between keyframes.
  - **STEP** Step – No interpolation between keyframes.
- **export\_pointer\_animation** (*boolean, (optional)*) – Export Animation Pointer (Experimental), Export material, Light & Camera animation as Animation Pointer. Available only for baked animation mode 'NLA Tracks' and 'Scene'
- **export\_animation\_mode** (*enum in ['ACTIONS', 'ACTIVE\_ACTIONS', 'BROADCAST', 'NLA\_TRACKS', 'SCENE'], (optional)*) – Animation Mode, Export Animation mode
  - **ACTIONS** Actions – Export actions (actives and on NLA tracks) as separate animations.
  - **ACTIVE\_ACTIONS** Active actions merged – All the currently assigned actions become one glTF animation.
  - **BROADCAST** Broadcast actions – Broadcast all compatible actions to all objects. Animated objects will get all actions compatible with them, others will get no animation at all.
  - **NLA\_TRACKS** NLA Tracks – Export individual NLA Tracks as separate animation.
  - **SCENE** Scene – Export baked scene as a single animation.
- **export\_nla\_strips\_merged\_animation\_name** (*string, (optional, never None)*) – Merged Animation Name, Name of single glTF animation to be exported
- **export\_def\_bones** (*boolean, (optional)*) – Export Deformation Bones Only, Export Deformation bones only
- **export\_hierarchy\_flatten\_bones** (*boolean, (optional)*) – Flatten Bone Hierarchy, Flatten Bone Hierarchy. Useful in case of non decomposable transformation matrix
- **export\_hierarchy\_flatten\_objs** (*boolean, (optional)*) – Flatten Object Hierarchy, Flatten Object Hierarchy. Useful in case of non decomposable transformation matrix
- **export\_armature\_object\_remove** (*boolean, (optional)*) – Remove Armature Object, Remove Armature object if possible. If Armature has multiple root bones, object will not be removed
- **export\_leaf\_bone** (*boolean, (optional)*) – Add Leaf Bones, Append a final bone to the end of each chain to specify last bone length (use this when you intend to edit the armature from exported data)
- **export\_optimize\_animation\_size** (*boolean, (optional)*) – Optimize Animation Size, Reduce exported file size by removing duplicate keyframes
- **export\_optimize\_animation\_keep\_anim\_armature** (*boolean, (optional)*) – Force Keeping Channels for Bones, If all keyframes are identical in a rig, force keeping the minimal animation. When off, all possible channels for the bones will be exported, even if empty (minimal animation, 2 keyframes)
- **export\_optimize\_animation\_keep\_anim\_object** (*boolean, (optional)*) – Force Keeping Channel for Objects, If all keyframes are identical for object transformations, force keeping the minimal animation
- **export\_optimize\_disable\_viewport** (*boolean, (optional)*) – Disable Viewport for Other Objects, When exporting animations, disable viewport for other objects, for performance
- **export\_negative\_frame** (*enum in ['SLIDE', 'CROP'], (optional)*) – Negative Frames, Negative Frames are slid or cropped
  - **SLIDE** Slide – Slide animation to start at frame 0.
  - **CROP** Crop – Keep only frames above frame 0.

- **export\_anim\_slide\_to\_zero** (*boolean, (optional)*) – Set All glTF Animation Starting at 0, Set all glTF animation starting at 0.0s. Can be useful for looping animations
- **export\_bake\_animation** (*boolean, (optional)*) – Bake All Objects Animations, Force exporting animation on every object. Can be useful when using constraints or driver. Also useful when exporting only selection
- **export\_merge\_animation** (*enum in ['NLA\_TRACK', 'ACTION', 'NONE'], (optional)*) – Merge Animation, Merge Animations
  - `NLA_TRACK` NLA Track Names – Merge by NLA Track Names.
  - `ACTION` Actions – Merge by Actions.
  - `NONE` No Merge – Do Not Merge Animations.
- **export\_anim\_single\_armature** (*boolean, (optional)*) – Export all Armature Actions, Export all actions, bound to a single armature. WARNING: Option does not support exports including multiple armatures
- **export\_reset\_pose\_bones** (*boolean, (optional)*) – Reset Pose Bones Between Actions, Reset pose bones between each action exported. This is needed when some bones are not keyed on some animations
- **export\_current\_frame** (*boolean, (optional)*) – Use Current Frame as Object Rest Transformations, Export the scene in the current animation frame. When off, frame 0 is used as rest transformations for objects
- **export\_rest\_position\_armature** (*boolean, (optional)*) – Use Rest Position Armature, Export armatures using rest position as joints' rest pose. When off, current frame pose is used as rest pose
- **export\_anim\_scene\_split\_object** (*boolean, (optional)*) – Split Animation by Object, Export Scene as seen in Viewport, But split animation by Object
- **export\_skins** (*boolean, (optional)*) – Skinning, Export skinning (armature) data
- **export\_influence\_nb** (*int in [1, inf], (optional)*) – Bone Influences, Choose how many Bone influences to export
- **export\_all\_influences** (*boolean, (optional)*) – Include All Bone Influences, Allow export of all joint vertex influences. Models may appear incorrectly in many viewers
- **export\_morph** (*boolean, (optional)*) – Shape Keys, Export shape keys (morph targets)
- **export\_morph\_normal** (*boolean, (optional)*) – Shape Key Normals, Export vertex normals with shape keys (morph targets)
- **export\_morph\_tangent** (*boolean, (optional)*) – Shape Key Tangents, Export vertex tangents with shape keys (morph targets)
- **export\_morph\_animation** (*boolean, (optional)*) – Shape Key Animations, Export shape keys animations (morph targets)
- **export\_morph\_reset\_sk\_data** (*boolean, (optional)*) – Reset Shape Keys Between Actions, Reset shape keys between each action exported. This is needed when some SK channels are not keyed on some animations
- **export\_lights** (*boolean, (optional)*) – Punctual Lights, Export directional, point, and spot lights. Uses "KHR\_lights\_punctual" glTF extension
- **export\_try\_sparse\_sk** (*boolean, (optional)*) – Use Sparse Accessor if Better, Try using Sparse Accessor if it saves space
- **export\_try\_omit\_sparse\_sk** (*boolean, (optional)*) – Omitting Sparse Accessor if Data is Empty, Omitting Sparse Accessor if data is empty
- **export\_gpu\_instances** (*boolean, (optional)*) – GPU Instances, Export using EXT\_mesh\_gpu\_instancing. Limited to children of a given Empty. Multiple materials might be omitted
- **export\_action\_filter** (*boolean, (optional)*) – Filter Actions, Filter Actions to be exported
- **export\_convert\_animation\_pointer** (*boolean, (optional)*) – Convert TRS/Weights to Animation Pointer, Export TRS and weights as Animation Pointer. Using KHR\_animation\_pointer extension
- **export\_nla\_strips** (*boolean, (optional)*) – Group by NLA Track, When on, multiple actions become part of the same glTF animation if they're pushed onto NLA tracks with the same name. When off, all the currently assigned actions become one glTF animation
- **export\_original\_specular** (*boolean, (optional)*) – Export Original PBR Specular, Export original glTF PBR Specular, instead of Blender Principled Shader Specular
- **will\_save\_settings** (*boolean, (optional)*) – Remember Export Settings, Store glTF export settings in the Blender project
- **export\_hierarchy\_full\_collections** (*boolean, (optional)*) – Full Collection Hierarchy, Export full hierarchy, including intermediate collections
- **export\_extra\_animations** (*boolean, (optional)*) – Prepare Extra Animations, Export additional animations. This feature is not standard and needs an external extension to be included in the glTF file
- **export\_loglevel** (*int in [-inf, inf], (optional)*) – Log Level, Log Level
- **filter\_glob** (*string, (optional, never None)*) – filter\_glob

FILE:

[addons\\_core/io\\_scene\\_gltf2/\\_init\\_.py:1071](#)

[Previous](#)  
[Export Anim Operators](#)  
[Report issue on this page](#)

Copyright © Blender Authors  
Made with [Furo](#)

[Next](#)  
[Extensions Operators](#)