COMPLETE USER GUIDE

UO Asset Studio

From your first profile to a finished release.

Version 0.31.0 | Windows desktop edition

Find | Prepare | Preview | Review | Save | Verify

33 chapters covering every Studio workspace and tool.

Step-by-step workflows, controls, recovery and troubleshooting.

Contents

Use the linked chapter titles or PDF bookmarks to jump to a tool. The offline HTML edition also filters chapters by search text.

About this guide

Complete user guide | Version 0.31.0 | Windows desktop edition

Browse, create, test and distribute UO content, with new-script generators for Sphere, RunUO, ServUO and ModernUO.

This manual describes the released 0.31.0 interface. It covers the main views, importers, workshops, House Builder, World Generator, file tools and recovery workflows. Button names match the application; a trailing `...` means that a button opens another window. Examples use generic names and do not identify free IDs on your installation.

The HTML edition works offline and includes a searchable contents panel. The PDF has a linked contents page and bookmarks. Keep the Markdown edition if you want to adapt the documentation for your own distribution. No game files, accounts, profile settings or private test projects are included.

01. Start here

What Studio does

UO Asset Studio connects your Sphere scripts to your Ultima Online client assets. You can find a definition, inspect its artwork and references, edit its script, import replacement or new assets, preview them, and review the files that will change before saving.

Studio is a native desktop application. Normal editing does not require a browser or a running server. Its Test Scene and House Builder walk test run inside the application. They show visual appearance and local movement; they do not execute your Sphere scripts.

The four things to keep separate

Thing	What it contains	How you preserve it
Working game files	Client art, animations, metadata and server scripts	A reviewed file save; a retained backup allows restoration
Editing session	Script drafts, open script-workspace documents and reviewed pending changes	Automatic checkpoints, or Asset workspace > Checkpoint now
Editable project	A house design, gump design or animation sequence	That workshop's Save project, Save design or Export edited sequence
Distribution	The content or documentation you intend to share	A deliberately prepared pack, export or distribution folder

A session checkpoint does not save game files. A screenshot does not save an editable design. A profile does not make an independent copy of your client or server.

A first session in ten steps

1. Install Studio, then open it from the Start menu.
2. Link an extracted Sphere server folder and the matching UO client-data folder.
3. Give the profile a clear name, such as `Development shard`.
4. If you need a separate working copy, use All tools > Create test copy and select the resulting profile.
5. Search for a familiar item in the definition list.
6. Select it and inspect Definition, Sphere scripts and Item artwork.
7. Open Test Scene to see the asset against a floor and walls.
8. Make a small change using the appropriate editor or importer.
9. Read the save review, including destination IDs, file formats, scripts and backup choice.
10. Save, then verify the result with the client and server that will actually use the files.

For a first experiment, copy existing equipment or add an item that reuses existing artwork. A new creature, wearable or house has more connected parts.

Reading this manual

Each chapter explains where a tool lives, how to use it and what is saved. Chapters 3 and 4 explain saving and recovery and apply throughout the manual. The task finder near the end helps you choose a tool. Troubleshooting covers common errors without asking you to start over blindly.

Recent additions

Version 0.31 adds reusable script recipes and a source-code preview to Generate server script. World Generator adds Height, Slope and Shoreline / blends views, a Terrain review tab with clickable inspection coordinates, and more reliable paths between reachable clearings. Classic UO mountain edges add bounded rock faces and textured slopes. Natural caves use black native cave mouths, and Rideable object mounts return their inventory item on dismount. Chapters 10, 12 and 32 explain these controls.

Version 0.30 opens on Home, with task cards, a permanent navigation sidebar and a searchable All tools page. Ctrl+K finds any tool; the active editor has optional Definitions and Script files panels. Appearance offers ten themes and two text sizes. Chapter 3 explains the layout.

Map Regions edits existing Sphere AREADEF / ROOMDEF boundaries and creates or removes regions. Load a read-only terrain map, draw or move rectangles, choose guarded, spell and travel rules, and review the script changes. Chapter 33 covers this new editor.

Import static walls, floors and items with Open folder or Open ZIP. Artwork and slots holds the source list and a larger preview; Properties and animation holds TileData, replacement and animation controls. Optional subfolders, naturally sorted previews and automatic free-slot selection make image packs quicker to bring in. Chapter 10 explains the limits and save review.

ModernUO is available as a preview C# target for official 0.15.6.198. It uses native generated serialization and can locate the UOContent project from the server checkout. Chapter 10 explains its destination and verification limits.

World generation now uses smoother biome boundaries, distance-based drainage and tree spacing that accounts for large canopies. Find area jumps to coast, forest, desert, snow, mountain, volcano or path previews. Changed settings are marked clearly and must be regenerated before export.

Choose Sphere, RunUO, ServUO or ModernUO on the Server script tab when importing assets, or generate a script for existing artwork through All tools > Generate server script. Complete armor sets have one target for the entire set. RunUO and ServUO both use C#, with separate templates for their different APIs. Chapter 10 explains the choices, destinations and limits.

Monster and Mount import now include Creature behaviour: eight editable presets, spells and healing, tameability and food, combat skills and resistances, attack FX and temporary summoned allies. Triggered abilities have a chance, cooldown, health threshold and range. Chapter 11 explains the controls and generated scripts; rideable objects retain their separate behaviour.

World Generator now includes Explore world in a new window. Pan and zoom through the generated landscape, use its minimap to travel, and jump to volcanoes, lakes and clearings. Chapter 32 explains the controls.

Attach sound now shows Current sounds, including resolved creature inheritance and detected sound commands. Listen to the current audio, choose a replacement, and review the change without adding a second sound command. See chapter 20.

Attach sound... links an existing client sound or ready-made PCM WAV to an item, creature, mount or script function. Choose when it plays, listen to it and review the script change before saving. Chapter 20 explains the complete workflow.

Workspace now has a visible Sound in UO panel above its preview. Choose Attach sound to this asset..., then use Save to UO files in the review. Chapter 26 also covers the new MUL to UOP direction, available beside UOP to MUL in the converter.

Save safety checks now test required file access before a reviewed save creates backups or starts preparing output. A second check runs before applying the changes. Direct script saves and restoration also check their destinations. Chapter 4 explains the checks and how to respond when a file is in use.

02. Installation, updates and profiles

Install or upgrade

Run the supplied Setup executable. The installed app includes its runtime; recipients do not need to install Python, Pillow, Inno Setup or developer tools.

To upgrade, save your work and close Studio, then run the new Setup. It detects the earlier installation and reuses its installation location. The confirmation step offers Upgrade, or Repair when installing the same version. Uninstalling first is normally unnecessary.

The default installation location is `%LOCALAPPDATA%\Programs\UO Asset Studio`. An older installation under Program Files may need Windows administrator permission to replace its application files. Use the same Windows account for that upgrade. The installer does not replace your linked shard/client data or saved profiles.

Keep `UOAssetStudio.exe` with its `runtime` folder. The `docs` folder contains documentation; `licenses` contains dependency notices. Do not keep your artwork projects inside `runtime`.

First-run project setup

Choose the folder containing your Sphere server and its `scripts` directory. A server ZIP is supported for browsing, but its scripts are read-only: extract it before editing or generating scripts.

Choose the UO data folder containing the actual client files, such as TileData, art and animation archives. The optional ClassicUO executable path is a remembered location; it is not required for Studio's in-app previews. Saving a profile does not configure a running game client to use those data files.

Click Save and Open Editor. The definition list comes from the selected server; asset previews come from the selected client folder. Use Item artwork > Check game files if you suspect that Sphere is reading a different folder.

Profile controls

Control	Use
Project selector	Switch to an existing profile
Manage > New profile...	Name a profile and select its locations
Manage > Edit paths...	Correct or change the active profile's linked locations
Manage > Rename...	Change the profile's display name
Manage > Save as new profile...	Copy the profile settings under another name
Manage > Remove profile...	Remove the saved profile entry; it does not delete the linked shard
Manage > Linked folders...	See the active locations and open their folders

Two profiles pointing to the same client folder edit the same client files. Use Create test copy, or prepare separate folders yourself, for independence. The last active profile opens next time. Finish open dialogs and saves before switching profiles; resolve any prompt about drafts or pending changes.

Settings and moving computers

Profiles are normally stored in `%APPDATA%\UOAssetStudio\project.json`. Back up this file if you want to preserve the profile list. On another computer, use Edit paths to update drive letters and locations. Settings are retained across application upgrades and uninstalling.

Advanced users can launch with `--config "D:\StudioSettings\project.json"` to use another settings file. That is optional; normal recipients can use the default. If settings are corrupt or another Studio instance has saved a newer copy, follow the reported conflict message instead of overwriting the file manually.

03. The main window and daily navigation

Home and the navigation sidebar

Home opens with your project status and six starting points: Creature or monster, Complete armor set, Static tiles from folder / ZIP, Generate server script, House Builder and World Generator. Explore all 26 tools opens the complete collection.

The permanent sidebar opens Home, All tools, Asset workspace, Definitions, Sphere scripts, Item artwork, Animations, Gumps, Hues, Asset spaces and Test Scene. Map Regions opens its own in-app window. Appearance is at the bottom. The project selector and Manage menu stay in the top bar.

Find a tool and keep the editor clear

Press Ctrl+K or choose Find a tool. Search by a task such as armor, sound, map or save, or choose Import, Assets, Scripts, Worlds or Files. Enter opens the first matching result. Ctrl+H returns Home. Alt+Left / Right move through your view history.

Definitions and Asset workspace initially show the definition browser; Sphere scripts initially shows Script files. Other views use the full editing width. Use the Definitions and Script files switches above an editor to choose which side panels are visible. The app remembers the choice for each view and keeps your open script text when switching views.

Drag dividers to resize the visible panels. Restored widths are checked so an editor or side panel is not left collapsed. Appearance > Reset panel layout restores the defaults. The minimum main-window size is 1100 x 720; some editors have their own minimum sizes or scrollbars.

Save All and Pending changes remain in the toolbar. The menu beside Save All contains Save scripts only, Save game files only and Save & recovery centre. Manage contains New profile, Edit paths, Linked folders, Rename, Save as new profile and Remove profile.

Search definitions

1. Enter a name, DEFNAME, identifier, path fragment or description in Search Sphere definitions.
2. Use the type selector to narrow the results to items or creatures.
3. Choose or type a Category / subsection. Empty search text browses the chosen category.
4. Select a row to populate the definition and script views.
5. Use Preview in Test Scene to open the selected content in the local scene.

All categories removes the category restriction. Categories come from the shard's actual CATEGORY and SUBSECTION fields. Definitions without those fields are still searchable. Numeric IDs and named identifiers are different ways to locate content; read the result's type before interpreting its number.

Definition and Workspace

Definition shows the selected section's properties, artwork and indexed references. Trigger assignments remain in the script; they are not all treated as top-level static properties. Double-click a reference to follow a resolved target. Back and Forward return through navigation history.

Workspace collects the current asset's related resources, save/session status and favourite/recent selections. Preview opens the relevant asset view. Find references starts the broader reference scan. Favourite toggles the current asset in your saved list. Detach preview opens its image separately. Export PNG saves the available image. Double-click related resources to navigate where supported.

Selecting a creature identifies a body animation; it does not turn that body number into an item-art ID. Use Selected creature in Animations when you want to follow a CHARDEF.

Themes

Appearance and the Theme menu offer Classic Studio, Invicta, Obsidian, Midnight Blue, Forest, Crimson, Royal Violet, Copper, Parchment and Arctic. Choose one to recolour the interface immediately. The selection is remembered per project. Appearance also offers Standard or Larger text and Reset panel layout. Themes do not recolour the original artwork or replace Test Scene's lighting controls.

Shortcuts to common tools

All tools provides the complete searchable tool collection. The Studio tools menu groups the same actions by category. Right-click menus provide actions for the item, gump, animation or script you clicked. Select the intended asset before using an action based on the active selection.

Select a definition, then use Asset workspace > Sound in UO > Attach sound to this asset... to link audio to the displayed item or creature. This panel sits above the large preview, so its button is immediately visible. It follows the asset shown in Workspace even if you later open another script. If you browse a raw asset without a linked definition, select its item or creature definition first. Right-click > Attach sound... remains available in the definitions list. For an effect function, open its script and use All tools > Attach sound.... Chapter 20 explains the sound controls.

Drag-and-drop supports a project script, an exported animation body folder, or image files. A single image dropped over an active item/gump can open replacement of that ID. An image batch offers item/static import or animation-frame import. A body folder opens the animation workshop for destination setup. A dropped script must belong to the active project. Use the normal file buttons when drag-and-drop is unavailable.

04. Saving, queues, backups and recovery

Choose the correct save command

Command	What it does
Validate and Save in Sphere scripts	Writes the displayed section or opened script document to its original .scp file
Save scripts	Saves the current script, or reviews changed script-workspace documents when applicable
Save game files	Saves the reviewed pending operations
Save All	Combines pending operations with changed script text and script-workspace tabs
Queue changes in a review	Adds that reviewed snapshot to Pending changes; does not yet write game files
Save original files in a review	Writes that reviewed operation immediately
Save project / Save design	Saves the workshop's editable document, separate from game files

Before writing client/server files, close the game, Sphere and other editors using those files. Reopen or reload the relevant client/server after saving so it reads the changes. Studio saving a script does not execute that script on a running server.

Automatic checks before saving

For a reviewed game-file save, Studio first checks every destination and the other native client archives and mappings in the selected client-data folder. On Windows, it tests exclusive access to those files and the permissions needed to write the destinations and create save files in their folders. Even an open reader that normally permits shared writing can stop the check.

All initial checks finish before Studio creates its save lock, backup, recovery journal or staged output. If a file is locked, missing unexpectedly, read-only or inaccessible, the Save safety check lists the affected paths and stops the operation. An initial failed check makes no backup or partial game-file change.

1. Read the reported filenames and reason for each failure.
2. Close the game, Sphere or editor using those files.
3. If access is denied, check that the active profile points to the intended writable working folder and that your Windows account has access.
4. Retry the save. If another editor changed the files, refresh and review your changes again when prompted.

The file-access check repeats after output preparation, immediately before the first replacement. Direct Sphere scripts saves check the target script before making a backup, and again before replacing it. Recovery and restore check their destinations before restoring any file. Existing conflict checks and recovery protection still apply.

Keep the relevant programs closed until saving finishes. A file-access check cannot detect an application that has already read its data and closed its file handles, and another program could open a file after the check. Studio does not close applications or change permissions for you. These checks do not make a multi-file save indivisible or replace retained backups.

Read a save review

Read the complete list of IDs and affected files. Where present, compare original and converted artwork, listen to converted audio, inspect frame coverage, and read the generated script. Check the chosen MUL/UOP destination and any linked mapping or resource-table changes. Back returns to the originating editor without saving.

Keep a backup is optional and normally off. Turn it on before the save if you want retained originals for later restoration. Removal operations keep a backup automatically. A save journal used to recover an interrupted operation is not a substitute for deliberately retaining backups of completed saves.

Work with the queue

Open Pending changes to inspect queued operations. Remove selected drops an operation from the queue; it does not undo an already completed file save. Use the backup checkbox in the pending-save window to retain originals for the combined save.

The queue holds a snapshot. Changing an import form or script after queuing it does not update that snapshot. If the new draft conflicts with the queued one, remove the old queue entry, review the current draft and queue it again. Identical script drafts can be recognised by Save All; different edits to the same binary record are not merged by guessing.

Separate operations can share an archive when they change different records. Automatic allocation considers supported pending imports, so their destination IDs are reserved during subsequent allocation. This is a local

planning aid, not a reservation across another program or another person's edits. Destinations are checked again when saving.

Script backups and manual backups

Sphere scripts has Back up original and an optional Also create a backup when saving checkbox. The script directory also offers Back up before opening for edit. These controls concern script-file backups. Back up original copies the on-disk file, not unsaved text in the editor.

Reviewed client-file backups live in `.uopve-save-...` folders beneath the client-data folder. Script-editor backups use `.uopve-backups`. Keep a reviewed-save backup folder intact, including its journal and linked originals.

Recover or restore files

1. Open Asset workspace > Save / recovery centre, or Item artwork > Recover client save.
2. For a failed or interrupted operation, choose Recover interrupted save.
3. For an intentional rollback, select a retained completed backup and choose Restore selected backup.
4. Save or discard current drafts and pending edits as directed, and close programs using the files.
5. Read the operation details and confirm restoration.
6. Reload the project and inspect the restored content.

Restoration applies to the complete linked operation: client assets, scripts and mappings. A new script created by that operation can be removed during restoration. If a file has changed again since the backed-up save, restoration can stop to protect the later work. Do not force an old entire archive over newer content; resolve which changes to retain first.

Restore an editing session

Asset workspace > Checkpoint now records drafts and the reviewed queue. Normal editing also schedules checkpoints. After reopening the same project, Restore session can recover them. The recovery centre also offers Open archived session and Discard saved session. Discarding a stored session does not erase game files or original-file backups.

Session queues depend on the files they were reviewed against. When those files change, Studio can retain draft text while requiring imports to be reviewed again. Sessions belong to the relevant project locations; they are not portable content packs.

Save unfinished animation edits with Export edited sequence and unfinished gump layouts with Save design. House Builder has its own Save project and automatic recovery. Do not assume every open workshop form or Test Scene arrangement is included in a general session checkpoint.

05. Sphere scripts and the script directory

Edit one definition

Select an item or creature and open Sphere scripts. This shows the chosen section, not necessarily the whole source file. Make the edit, then use Validate and Save or Ctrl+S. The source path tells you which file will be updated.

Validation checks the section boundary, changed headers, accidental additional sections, supported control-block structure and whether the original has changed on disk. Warnings about legacy syntax need interpretation; Studio is not the Sphere interpreter. Keep unrelated sections out of a section-only edit.

Reload reads the current disk content. Resolve an unsaved-edit prompt before discarding work. Script editing preserves supported encoding and line-ending information. If a file has been changed by another editor, reload and reapply the intended change rather than overwriting the newer version.

Find text in the displayed script

Ctrl+F focuses Find. Type literal text; matching spans are highlighted and the count is shown. F3 moves forward, Shift+F3 backward. The arrow buttons do the same. Aa makes the search case-sensitive. Clear removes the search term. In the Find entry, Enter and Shift+Enter navigate; Escape returns focus to the editor.

This search covers the displayed text. To search other files, use the script workspace's Find / replace across files.

Open and create files

The right-hand Script directory follows the active profile. Double-click a .scp or select it and click Open file to edit the complete document. Refresh updates the tree. The New button or right-click menu offers New folder and New .scp file under the selected location.

Use clear filenames and categories. Sphere resource filenames must remain unique across the script tree for generated content. Creating a file in an arbitrary folder is not the same as adding it to Sphere's resource loading configuration; confirm that your server loads it. Studio-generated imports register their new script when needed.

Delete a script deliberately

Right-click a .scp and choose Delete script to review deletion of the whole file. Read its contents and the removal list before choosing Remove listed content. This operation keeps a backup. It retains other scripts and client assets; references elsewhere may need updating.

Remove empty .scp file is a separate action for a script with no remaining meaningful content. Use it after inspecting a leftover file. If your aim is to remove an item/creature and its assets, use Remove related content instead; chapter 24 explains its scopes and limits.

Work with multiple script tabs

Open All tools > Sphere script workspace. Open script selects a file inside the active server. Open current brings in the current script. Open more files to work in tabs; Close tab closes the selected document after handling its edits.

Ctrl+Space offers definition and Sphere-keyword completion. F12 or Go to definition follows a supported symbol; choose the correct target if there is more than one. Find references searches for the relevant symbol. Completion and reference navigation help with source editing but do not prove that code will run correctly.

Use Review save all to inspect changed workspace documents together. The main Save All also includes changed workspace tabs. A reviewed or queued save leaves this workspace open; check the queue if an older draft conflicts with your current text.

Replace text across selected files

1. Open Find / replace across files in the script workspace.
2. Enter Find literal text and Replace with. This is literal, case-sensitive replacement, not regular-expression matching.
3. Set Path filter, for example `scripts/**/*.scp`, or narrow it to the relevant folder.
4. Click Find changes. Select a result to inspect its before/after difference.
5. Select only the files you intend to change; Ctrl/Shift selects multiple rows.
6. Click Review selected replacements and read the complete proposed changes.
7. Queue or save the reviewed operation.

This tool reads saved script files. Save or reconcile any open drafts for those files first. A matching word can occur in comments or unrelated contexts, so inspect each selected file rather than treating an occurrence count as approval.

06. IDs, file formats and artwork preparation

Identify the kind of number

Number	Meaning
Item art ID	Ground/backpack artwork; its TileData record contains item metadata
Body ID	A creature, human or equipment animation body
Raw body and MUL bank	Physical animation destination within a particular bank
Sphere body ID	The body number used by the server, potentially mapped to another bank
Gump ID	Interface or paperdoll artwork
Hue ID	A colour table; game hue 0 means original colours
Multi ID	A compiled house/structure layout
DEFNAME	A named Sphere identifier, such as i_example or c_example

These are separate namespaces. Item 400 and body 400 are not interchangeable. Prefer explicit `0x` prefixes when entering hexadecimal. Fields labelled hex can interpret bare input differently from fields labelled decimal. `0x0190` and decimal `400` describe the same number, but are examples rather than suggested destinations.

MUL, UOP and Auto

MUL assets commonly use a data file plus a matching index, for example art.mul with artidx.mul. UOP stores entries inside an archive. Auto selects the format supported by that tool and found in the chosen client folder; it is not an instruction to update every format present.

Item artwork Auto normally prefers ArtLegacyMUL.uop when present. Animation Auto follows supported client mappings and available sources. Use an explicit source to investigate a mismatch. House compilation is a special case: it updates both MUL and UOP multi representations when both are present.

Editing a MUL while your client reads the corresponding UOP can leave the game looking unchanged. Check the review's affected files and the actual data location used by the recipient's client. Loose overrides, Verdata and client-specific mappings may also take precedence.

Prepare images and animation files

Use PNG/BMP for still art and supported image sequences; use GIF where the importer offers it. Preserve transparency and avoid painting a checkerboard into the image. Native UO artwork uses reduced colour precision and, in these import workflows, binary transparency. Inspect the converted preview: it is more relevant than the original full-colour source.

Keep animation frame canvases and ground origins consistent. A body VD contains supported action/direction data and palettes; a GIF or PNG sequence supplies frames, not all the behaviours of a complete creature. GIF timing does not become a MUL animation timing field.

Wearables need an inventory icon, equipped movement animation and paperdoll images. Paperdoll pieces must retain the complete intended canvas and alignment. Cropping the paperdoll tightly around the clothing can move it to the wrong place on the body.

Names and capacities

Client names often have byte limits, not unlimited text length: TileData and hue names use 20-byte fields; sound-import names have a 39-byte Windows-1252 limit. Use short supported names when a converter reports an encoding or length error.

Most formats use the allocation already present in your data. Equipment import has a specific opt-in expansion option, described in chapter 13. It does not make every archive or metadata table unlimited. A free candidate is evidence from the checked files, not a global promise that no world object or dynamic script uses that number.

07. Item artwork, TileData and the paperdoll

Inspect an item

Open Item artwork. Enter an Art ID (hex) and click Inspect, or select a Sphere item and click Use selected item. The latter follows supported item aliases, such as ID/DUPEITEM links. The view shows artwork and the client TileData record.

Export PNG saves the available item image. Copy art ID copies its number for another tool. Browse slots opens the allocation browser. Right-click the image for replacement, Show in Test Scene, Open asset workspace, Find references, Favourite / unfavourite, Export PNG or Detach preview.

Client metadata and Sphere properties are different layers. For example, a client tile's displayed name or flags do not establish every gameplay property of a scripted item. Check both Item artwork and Definition/Sphere scripts.

Edit TileData

1. Inspect the correct item ID and choose Edit TileData.
2. In Item properties, adjust the fields you understand: name, weight, height, quality/equipment layer and animation ID.
3. Open Choose TileData flags to inspect named flags and descriptions.
4. Use the other stored-fields tab only when you need the specific raw value; preserve unknown fields.
5. Review changes, checking the ID and changed fields, then queue or save.

The flag picker supports name search, descriptions, double-click or Space to toggle, a full hexadecimal mask and additive presets. Presets include equipment, containers, surfaces, blocking decorations and animated statics. Additive means existing bits stay set: inspect the final mask and remove incompatible flags explicitly.

Weight 255 has special immovable conventions. Height affects geometry; equipment layer and animation affect wearing. Do not convert a decorative item into working equipment by changing only one field. Use the equipment importer when creating a complete wearable.

The small paperdoll

For supported equippable items, Item artwork shows Equipped item / paperdoll. Choose Male or Female and enter a hue; press Enter to refresh where required. This uses the linked paperdoll art, not the backpack icon.

Wear in Test Scene loads the selected wearable on the human model and shows the paperdoll alongside it. The paperdoll is a one-item inspection tool. Use Equipment / mount preview for manually combining multiple animation layers.

If the paperdoll is missing, inspect the item's equipment animation link and the corresponding male/female gumps. For the standard equipment link, male and female gump IDs are animation ID + 50000 and + 60000. These are linked destinations, not arbitrary empty gump slots. Queued supported TileData/gump changes can be included in the preview, so refresh after changing the queue.

08. Asset spaces and choosing destinations

Asset spaces is the gallery for item and animation allocations. Choose Items or Animations, then a source. Reload reads the current files. Search by name or ID and use the All, Free candidates, Used or applicable No UOP entry filter.

Green cards are free candidates; gold indicates used/referenced content. Double-click a card or use Open details to inspect it. Import into slot opens the relevant importer with that destination. Selecting a slot manually does not remove later conflict checks.

Animation galleries and UOP ranges

MUL animation galleries describe their bank's body allocations. Modern UOP bodies use hashed entries and are browsed in ranges. UOP start body and Go jump to a range; scrolling can extend the loaded inventory when the search is empty and All is selected. Filters describe the loaded range. No UOP entry does not mean the app can write a modern UOP body there.

Slot browsers inside importers

Browse slots and similar buttons open a more focused candidate list. Search ID / name / reason, choose All slots, Free candidates or Used slots, and move between pages. Use selected ID copies the destination into the importer. Read the reason a slot is considered occupied.

The allocation checks consider supported saved assets, client metadata, indexed script references and applicable pending imports. World saves, all map/override cases and runtime expressions are not universally covered. Reference scans add useful evidence before repurposing old content.

Automatic placement

Leave automatic placement enabled for new content. Find free space restores automatic choice after manual editing. Review the suggested range before saving:

- Separate item/gump image batches use a consecutive range in automatic mode.
- Animated statics need consecutive artwork IDs with their relative frame order preserved.
- Complete VDs need a compatible whole body block.
- Mounts also need riding-memory and, when importing a new icon, icon destinations.
- New equipment appearances need the animation and both paperdoll destinations together.

If there are free individual slots but no sufficiently long block, a consecutive batch still cannot fit. Reduce a separate-image batch or choose another supported source/bank. Do not split an animated-static sequence casually. Equipment capacity expansion only solves the specific animation-index shortage covered later.

09. Add item: reuse art or export an import kit

All tools > New Sphere item is useful for a basic static item. The Import tools can directly prepare client-file imports.

Reuse existing artwork

1. Choose Use existing artwork.
2. Enter the item name, a unique script identifier, artwork ID, category, subsection and weight.
3. Choose the script folder and a unique .scp filename.
4. If the art is newly imported and truly has no existing numeric Sphere base, use the corresponding numeric-base option; do not introduce a duplicate base.
5. Open Review item and inspect the preview and generated definition.
6. Choose Create script when correct. Studio writes and opens the new script.

This creates a basic item definition using the chosen graphic. It does not author creature animation or supply the movement/paperdoll parts of equipment. When the server already defines that art base, use inheritance rather than creating a competing numeric definition.

Prepare new PNG for UOFiddler

Choose that mode, select a single PNG, and enter a candidate art ID, TileData name, weight, height and flags such as Blocks movement or Surface. Review the converted image and script, then Export import kit to a separate output folder.

The kit contains the image, item metadata, script and instructions for the external import. Exporting it does not patch the client's art/TileData or install the new script. Complete the kit's import steps against a working copy before enabling its script on the target server.

Find a free candidate in this starter wizard has narrower coverage than every other Studio tool. Its MUL-based check needs matching art.mul, artidx.mul and tiledata.mul; UOP-containing setups may require an extracted MUL working copy. A kit does not reserve its candidate forever. Recheck it at import time.

For a fully in-app direct import, use All tools > Static tiles from folder / ZIP, described next. Keep the original PNG and kit if you intend to share or revise the artwork later.

10. Import items, animated statics and effects

Choose Sphere, RunUO, ServUO or ModernUO

The importers have a Server script tab with a Script target choice. Sphere generates .scp files for Sphere X. RunUO generates C# .cs files for the official RunUO 2.x source. ServUO generates C# .cs files for official pub57. RunUO and ServUO use the same language, but their APIs differ; choose the actual server so weapon callbacks and rideable objects use the correct template.

This choice covers Monster, Mount, Item, Effect, Clothing and Weapon imports. Complete armor sets have a Script target for entire set control in their main window, including pieces loaded together from a folder. Each piece keeps its own appearance links and gameplay settings.

Sphere uses your active project and registers its new file in spheretables.scp. For a C# target, choose the C# server / export folder: the root containing Scripts, or an empty directory for export. Studio creates the selected category folder under Scripts at save time. It uses the UOAS_ prefix for generated class names, rejects existing files and duplicate class names, and shows the destination and complete source in Review. C# files do not register in spheretables.scp.

Review, queueing, file-access checks and recovery cover the C# file together with any imported assets. Compile the result with the chosen server, restart as required and use the displayed GM creation command. Studio does not compile or start your server. Every player's compatible client also needs the matching assets and metadata.

For artwork already installed, open All tools > Generate server script (Sphere / RunUO / ServUO / ModernUO). Select the kind, enter the existing graphic/body ID, choose properties and a target, then Review script and destination. This creates a script without importing artwork. It still uses the working profile and linked client directory for its save journal. Use decimal IDs or an explicit 0x prefix.

Save a script recipe and preview the source

The standalone Generate server script window has Open recipe, Save recipe and Preview code above its tabs. Configure the asset, target, names, stats and rules, then choose Save recipe. Save the .uoscript.json outside your linked game folders. It keeps the graphic/body ID, mount IDs, equipment properties, bonuses and creature abilities that have been added to their lists. An unfinished rule still in its entry fields is not included; choose Add or Update first.

Open recipe restores those settings. Change the display name, identifier and artwork IDs when creating another asset. Recipe files contain data only, not executable scripts. Server/export paths, destination folders and filenames are excluded; choose the destination for the current project. Recipes are limited to 64 KB and unsupported or invalid settings are rejected before replacing your controls. These recipe buttons belong to the standalone generator, not the import wizards.

Preview code shows a read-only snapshot of the generated source and creation commands. Copy code copies that source. The preview does not save, compile or run it. Close the preview and choose Review script and destination when ready: the normal save review rechecks current files, duplicate names and the chosen destination. For Sphere, a loaded script project is needed to resolve existing definitions. For C# targets, native compilation and shard testing are still required.

ModernUO preview target

ModernUO also uses C#, but its constructors, combat callbacks and serialization differ from RunUO and ServUO. Select ModernUO in the same Script target control. The reference is official release 0.15.6.198.

Choose the ModernUO source checkout containing Projects/UOContent, or that UOContent folder itself. Studio creates category subfolders under Projects/UOContent/Scripts. An empty export folder also works: copy the generated Scripts folders into the recipient content project before compilation. Name checks cover the whole content project, including native Items and Mobiles, and exclude bin/obj output.

Generated ModernUO classes use Constructible, partial classes and native generated serialization. Creature cooldowns use anchored timestamps; summon references are saved and owned spell-target timers stop on deletion. Keep ModernUO-generated migration files with the server source and follow its migration procedure before changing a deployed class.

This is a preview target checked against official source. A complete ModernUO build and native gameplay/serialization tests were not run for this release because its required packages were not present in the existing build environment. Compile and test in a separate ModernUO project before deployment. Studio does not install or compile server dependencies. The existing Sphere-only tools described below retain their scope.

C# behaviour and compatibility

C# monsters use BaseCreature; animal mounts use BaseMount; rideable objects use a native ethereal-style inventory item without an animal to tame or feed. Stats, skills, resistances, tameability and follower slots are generated. Diet controls accepted FavoriteFood; None does not disable native pet-loyalty rules.

Normal selected spells use native mana, Magery, cast timing and targeting. During combat, the generated scheduler considers a spell at most every four seconds and prioritizes Cure and healing below the chosen threshold. Triggered abilities retain their chance, cooldown, health threshold, range and line-of-sight checks. During combat rolls follow the server's AI decision frequency.

Instant C# procs have explicit preset strength: power 500 gives 50 damage or healing before native mitigation. Poison level uses power / 250, capped at four; paralysis lasts at least one second using power / 100; stat debuffs use power / 100 for power / 10 seconds; Mana Drain removes power / 10 mana. Fixed and proc healing do not heal poisoned targets. These procs use no mana or reagents. Use normal spells for Protection, Bless, Reactive Armor and Magic Reflection. Existing Sphere functions cannot run on C# servers and are rejected.

Summons use an existing C# BaseCreature class such as Skeleton. Preset names are mapped automatically. Generated code tracks living limits, saves cooldowns and summon references, uses native summon expiry, and removes children when the parent dies or is deleted. Minions cannot be tamed and cost no follower slots; generated abilities cannot recursively summon. Native summoning removes carried loot. An empty export directory cannot check whether a custom class exists; the recipient's compiler must resolve it.

C# clothing uses BaseClothing; armor pieces use BaseArmor with generic plate material; weapons use the selected native weapon family. Bonuses use native attributes, resistance fields and up to five skill-bonus slots. Expansion rules determine their effect. Legacy weapon speed is converted to modern seconds as 150 / speed, with a 0.25-second minimum. Select Armor piece to use an armor rating.

C# effect imports create a GM-only preview item and a public Play(Mobile) method for custom scripts. They create visual effects, not a complete damaging spell. Animated artwork still depends on the client's animation metadata.

Tools that remain Sphere-specific

The profile and definition index, source editor, sound-link editor, related-definition removal, house/deed scripts, world entrance scripts and content-pack script remapping remain Sphere workflows. C# generation does not convert arbitrary existing scripts or make C# classes appear in the Sphere editor. Share generated .cs files and matching client IDs separately instead of remapping them through Sphere content packs. The following import chapters describe Sphere output unless they explicitly mention C#.

The templates were compiled against official RunUO 2.x and ServUO pub57 sources, with constructor and serialization checks. Different forks may need changes. Test Scene previews artwork; it does not execute server AI,

combat or equipment bonuses. Verify native spell scheduling, taming, mounting and balance on a working-copy shard. SERVER_SCRIPTS.md in the source package records detailed compatibility notes and exact reference revisions.

Choose the entry point

All tools > Static tiles from folder / ZIP prepares artwork and a matching Sphere ITEMDEF. All tools > Animated item or effect prepares artwork and a callable effect function as well as its item definition. Item artwork > Import images / animated item opens the art workflow when you mainly need client-file editing; inspect whether script generation is enabled on its Server script tab.

Import one item or several independent items

1. Click Choose images for PNG/BMP files, Open folder for a tile folder, or Open ZIP for an image archive. Select a row to inspect its converted preview.
2. Choose Art files: Auto, MUL or UOP.
3. Leave automatic placement enabled, or deliberately choose the first art ID and mapping mode.
4. Use Move up / Move down to establish import order. Read the destination list in the review.
5. Leave Make these frames one animated static item off for independent items.
6. Decide whether to Set TileData name, weight, height and flags for the listed IDs. Blank name uses the filename; other stored fields remain unchanged.
7. On Server script, enable generation if needed and set name, identifier, category, subsection, folder and filename.
8. Click Review files and script. Verify every proposed ID and definition before Queue changes or Save original files.

Consecutive IDs maps rows to a continuous range. IDs from filenames uses the file naming convention instead; use clear numeric filenames and inspect the proposed mapping. It switches away from automatic allocation. Do not assume a package's original reference IDs are vacant destinations on your shard.

Replacing existing artwork is an explicit choice. Enable Replace existing artwork at the listed IDs only when those are the assets you intend to change. For one replacement, the right-click Replace ... with image action is often clearer because it starts from an inspected ID and shows before/after art.

Import a static tile folder or ZIP

Open All tools > Import static tiles (folder / ZIP), or use the same Open folder / Open ZIP buttons in the item artwork importer. This imports static walls, floors and items, not ground terrain tiles. Choose Include subfolders before opening the source. PNG and BMP images are listed in natural filename order within their folders (tile2 before tile10). Other file types are skipped; ZIPs are read without extraction.

Each batch supports up to 256 still images, 1–512 pixels per dimension, 16 million pixels in total, and 128 MB of input. Each image file must be under the 16 MB limit. Corrupt images, unsafe or duplicate ZIP entries and password-protected archives stop the load without replacing your previous selection. Split large packs into smaller folders or ZIPs. Cancel stops the load without queuing any changes.

Folder names remain visible in the list. Blank TileData names use each image's filename without its folder or extension. IDs from filenames reads the final filename: for example, `walls/0x4000.png` requests item art ID 0x4000. Duplicate destination IDs are rejected even if the files are in different folders. Automatic placement instead finds a consecutive free range, taking queued changes into account.

For independent tiles, leave the animated-static option off. Set the appropriate TileData collision, surface, height and other flags; the importer cannot infer a wall's intended behaviour from its pixels. The chosen settings apply to the batch, so import walls and floors separately when their properties differ. Review every ID and proposed file change before saving. Loading a folder or ZIP alone changes no client files.

Make an animated static

An animated static is item art driven by animdata.mul, not a creature body animation. Examples include a decorative animated object or frames for a visual effect.

Choose a GIF or ordered image sequence and enable Make these frames one animated static item. Use 1-64 consecutive artwork IDs in frame order. Play frames checks order at the tool's preview rate. Set Frame interval to 1-255 client ticks. FrameStart accepts 0-255; it is an advanced stored field, not a general GIF start-time selector. Leave its default unless you are reproducing a known arrangement.

The reviewed save includes the relevant art, TileData and AnimData changes. An animated item gets its main script definition at the first artwork ID; independent images get separate definitions. The native AnimData sequence has a supported frame-count/offset limit, so a long GIF may need deliberate preparation rather than direct import of every frame. Studio reports unsupported sequences before saving.

Generate an effect script

Choose All tools > Animated item or effect. The art setup follows the animated-static workflow. On Server script, choose Follow character or Ground / object and set Effect speed and Loop duration. These are fields for the generated effect call; they are distinct from frame playback and image timing.

Studio generates `f_<identifier>` and the necessary item definition, using the first artwork ID. The review shows the actual script and test command; use that command on a suitable test shard after saving/loading the files. Test Scene can preview ground, on-player and projectile visuals, but it does not run the generated function.

The effect importer creates a visual effect. It does not automatically design damage, targeting rules, cooldowns, team filters or a complete spell. Add those server behaviours deliberately in Sphere scripts. If the art already has a numeric base, inspect the generated ID inheritance so the new named definition uses that base rather than duplicating it.

Generated script fields

Display name is the player-facing text. Identifier supplies the named script symbol; use letters, digits and underscores with a sensible prefix supplied by the importer. Category and Subsection organise the definition list. Script folder and New .scp filename identify the output. Scripts default to type-specific folders and are registered as resources when needed.

For monsters/mounts the same panel includes basic STR/hits, DEX and INT fields. For equipment, gameplay fields come from Item and icon. Review the full generated script rather than assuming a template fits every shard's custom systems.

11. Import monsters and other characters

Open All tools > Creature or monster. Despite the button name, the important distinction here is a character/body animation import. The animation layout describes how frames are stored; the generated CHARDEF supplies gameplay properties.

Choose a source and destination

Choose frames, GIF or .vd. PNG/BMP sequences and GIFs supply one action and stored direction. A compatible VD supplies its populated sequences and layout. Review the sequence list before choosing a destination.

Field	What to check
MUL bank	The physical animation files being written, banks 1 through 6
Raw body ID	The destination inside that bank
Action	The animation action for a frame-sequence import
Stored direction	One of 0 through 4; the remaining display directions use mirroring
Animation file layout	13 Animal, 22 Monster or 35 Human actions; VD determines its own format
Sphere body ID	The server-facing number; secondary banks may require linked mappings
Origin X / Origin Y	Frame registration for image imports; use the preview to align the ground

Keep automatic free-block selection on for new content. Browse body slots allows deliberate selection. Bank 1 normally uses the same raw and Sphere body ID; secondary-bank imports can prepare the necessary mappings with the generated script workflow. Confirm those changes in the review.

Import a complete body

1. Load the compatible VD.
2. Check its layout and populated action/direction sequences.
3. Choose a suitable bank and Find free space.
4. Preview several directions and actions, especially stand, walk, run and required combat/death actions.
5. Enable script generation and set a unique name, identifier and destination script.
6. Configure Creature behaviour, then review all native files, mappings, the CHARDEF and any ability helper functions.
7. Queue/save, then test the character on the target shard.

Import a single sequence

Choose the image sequence or GIF, assign the intended action and stored direction, and set a compatible body layout. Adjust origins, update the preview and reorder frames where necessary. Import other required directions/actions separately.

A walking GIF in one direction is not a complete monster. The template can create a basic walking monster definition with combat properties and a monster brain, but missing movement/attack/death art remains missing. Preview rate does not alter the server/client movement timing.

Creature behaviour: spells, taming and special attacks

Open Creature behaviour in a Monster or Mount import. Enable Generate and install on Server script so these choices are included in the new script. This configures a new imported creature; it does not edit existing creature definitions. STR / hits, DEX and INT / mana remain on Server script.

Apply preset offers Basic melee, Tameable animal, Armored fighter, Spellcaster, Self-healing fighter, Poisonous creature, Dragon-style pet and Lich-style summoner. Applying a preset replaces the current behaviour settings and ability list; caster presets also set INT to 100. Every value remains editable.

Page	Controls and purpose
Stats, taming & food	Animal or Monster brain; minimum/maximum damage, armor, fame and karma; Wrestling, Tactics, Magic resistance, Magery, Evaluating intelligence and Meditation; physical, fire, cold, poison and energy resistances
Taming and feeding	Can be tamed, required taming skill, follower slots and Meat, Plants, Omnivore or None diet. None disables hunger. Taming a monster does not add riding animations; use Mount import for a rideable creature
Spells & healing	Casts spells using normal Sphere AI; choose offensive, healing, cure and defensive spells. Set Heal at or below HP %. Magery must be above zero; mana, spell strength and normal casting are governed by the shard
Triggered abilities	Add ability, select/edit, Update selected, Remove selected and New / deselect. Up to 12 rules, each with its own trigger, chance, cooldown, own-health threshold and enemy range

During combat rolls when Sphere makes a combat decision, not at a fixed rate per second. On a landed hit fires after a successful melee hit check. When hurt fires during incoming damage handling, before that damage is subtracted from HP. Health thresholds always refer to the creature's own health. For example, HP at / below 40% makes a rule eligible only while it is at or below 40% health. Enemy abilities check distance and line of sight.

Ability	What it generates
Spell effect	An instant spell proc, such as Flamestrike, Fireball, Poison or Paralyze against the enemy, or a beneficial spell on self. Power 500 means 50.0 skill units. It uses no mana, reagents or casting delay
Heal self	Restores a fixed amount of the creature's HP, capped at maximum. Set a low-health threshold and cooldown. This uses no mana or bandages
Visual / damage effect	An existing client effect art ID, shown On target, as a Projectile or on the Ground. Optional sound ID and physical damage. Zero damage is cosmetic; self visuals require zero damage
Summon creatures	Existing CHARDEF such as c_skeleton, number per activation, maximum living summons and lifetime. Temporary allies belong to the summoner, cannot be tamed and expire automatically

Ability	What it generates
Existing function	A creature-compatible Sphere function selected from the linked server, such as a suitable f_fire_nova. It runs on the creature with the enemy as SRC and ARGN1; that function controls targeting, visuals and damage

To make a lich-style summoner, apply Lich-style summoner, open Triggered abilities, select the summon rule and adjust the creature, chance, cooldown, count, limit and lifetime. Choose Update selected to keep the edit. Stock lich behaviour differs between shard script sets; this preset is a starting point you control.

Each summon rule creates 1-5 creatures per activation, allows up to eight living children and gives them a 5-600 second lifetime. The combined living limits across rules cannot exceed 12 per summoner. Summons are removed when the summoner dies or is deleted, and do not consume follower slots. They retain the chosen existing creature's shard scripts, which may add other behaviours. Studio-generated triggered abilities are suppressed on generated minions to prevent recursive summoning.

Review files and script shows the complete CHARDEF and helper functions before saving. They stay together in the monster or mount script and through queued saves/checkpoints. Effect art IDs and sound IDs must already exist in the client; this page does not import separate effect artwork or audio. A listed function must be suitable for NPC use. Native spell selection, combat balance, resistance rules and mounted actions need testing on a working-copy shard: the in-app Test Scene previews artwork and movement without running Sphere combat AI.

Replace part of an existing body

Inspect the existing body and source first. Enable Replace the listed destination sequences only for an intentional update. Unlisted actions/directions remain unchanged; this is useful for targeted corrections but can also leave a body with mixed old/new artwork. Check the sequence list and the final body in multiple actions.

Original VD transfer preserves its supported palettes and origins. PNG conversion can requantise colour. Modern UOP body animation is browsed/exported, while this writer targets supported MUL sequences; it does not provide automatic full modern-UOP-to-MUL conversion.

12. Mounts, rideable objects and inventory icons

Open All tools > Mount or rideable object. A working mount combines body animation, server definition, riding-memory item/mapping and inventory/shrink artwork. Each part has its own role; an icon does not supply a mounted pose.

Import a mount

1. Choose mount .vd and inspect the supplied layout.
2. Select the MUL bank and leave Automatically use next free animation block and linked IDs enabled.
3. Find free space. Review raw/Sphere body, mount-memory and new icon destinations.
4. Set the memory item's short client name. Do not reuse a referenced memory record without deliberately enabling and reviewing replacement.
5. Choose the inventory image on Shrink icon.
6. On Server script choose Animal or Rideable object and complete the script fields.
7. Preview actions/directions and open Review files and script.
8. Save and use the generated test command to create the correct object on your test shard.

Compatible 35-action mounts retain the human-style layout and require the relevant unmounted walk/run/stand and mounted walk/run/stand sequences. For that layout, the required pose groups include actions 0/2/4 and 23/24/25 in five stored directions. Other supported layouts have their own checks. A layout named Human does not force the mount to behave like a human character.

Animal or Rideable object

Animal is the default template for a tameable mount. Rideable object is for a unicycle or contraption without taming/feeding. It still uses Sphere's character/mount machinery internally, with ownership and no autonomous wandering/following in the generated object workflow.

For a rideable object, use the generated `.add i_ride_...` command shown in the review. Put one item directly in the character's backpack and double-click it to mount immediately. Dismounting returns the same item to the backpack. Sphere hides the original item while ridden and removes its temporary backing creature on dismount; repeated clicks cannot create another mount. The item is blessed and keeps its hue. RunUO, ServUO and ModernUO use their native EtherealMount class. Test the complete cycle on your shard, especially custom death, trading and mounting hooks. Simply spawning the creature body bypasses the intended item workflow.

This choice generates the new script. It does not automatically convert earlier installed animal scripts or change an already deployed world object.

Choose a shrink or inventory icon

Shrink icon offers existing artwork, a new PNG/BMP, or a frame from the loaded mount animation. Existing artwork can be searched by name, DEFNAME, category or ID, with Figurines first / Figurines only / All artwork filters. Find horses helps locate existing horse-like figurines.

For a new image, choose a separate free art slot and a short client name. The riding-memory item and the new visible icon must not be the same destination. Reusing shared icon art does not mean you should overwrite its existing Sphere definition.

Turn a pose into the icon

Select Use animation frame, then Choose pose. Choose an action/direction and a frame thumbnail. Set Fit within pixels, Trim transparent edges and Mirror pose if required. Inspect the converted still image, then Use this frame as icon.

The selected frame becomes still item artwork saved with the mount; no intermediate PNG export is necessary. Changing the source VD clears the previous frame choice to avoid accidentally keeping an icon from another mount.

Check the completed mount

Test standing and travelling in every direction, the rider's seat, overlap with clothes and the inventory/shrink image. Studio's rider offsets are preview controls; changing them is not a saved animation correction. Use the animation tools or source artwork to make a persistent registration change. Ownership and feeding rules need server testing; Test Scene checks appearance only.

13. Clothing and weapons

Open All tools > Clothing or weapon. The five tabs are Item and icon, Equipped appearance, Animation preview, Bonuses and effects, and Server script. Use this workflow for one independently equippable piece.

Understand the required assets

Part	Purpose
Inventory icon	The item on the ground or in a backpack
Equipment animation	How the worn item looks on the moving world character
Male paperdoll image	The item overlaid on the male paperdoll
Female paperdoll image	The item overlaid on the female paperdoll
TileData and script	Links those assets to the layer and gameplay properties

One image cannot stand in for all of them. A whole-suit animation is one appearance; it cannot automatically be divided into a helmet, chest and boots that can be worn separately.

Item and icon

Choose existing equipment to copy when you want a starting point. Check the new destination item ID, short client name and Artwork files source. The copied appearance/icon is a source; the destination should still be deliberate.

Choose the Equipment layer and set weight, required strength and durability. Clothing has Armor, Armor piece (Sphere t_armor) and Dyeable in Sphere. Weapons have Weapon type, minimum/maximum damage and Sphere weapon speed. Partial hue limits colouring to grey pixels where supported; it is separate from the preview hue selector.

Icon source Import reads a PNG/BMP. Copy reads another item ID. Preview icon shows the backpack/ground artwork after conversion. Check that a helmet's icon is not accidentally being reused for every piece of a set.

Reuse an equipped appearance

On Equipped appearance, choose Reuse and select the existing equipment animation ID. Studio reads its animation and linked male/female gumps. This is useful when making a new scripted item with an established appearance.

Choose the paperdoll source matching the recipient's files. Load appearance and check action coverage, then inspect the male/female images. Reuse does not mean the existing shared animation is copied to a new body or rewritten.

Import a new equipped appearance

Choose Import. Supply a compatible 35-action human/equipment VD and both paperdoll PNG/BMP files. Preserve the full canvas and transparency of the paperdoll pieces. The equipment animation links to male gump `50000 + animation ID` and female gump `60000 + animation ID`.

Keep Next free animation and paperdoll bundle enabled, or use Free animation + gumps for an intentional destination. Both linked paperdoll slots must be available as well as the animation body. Load piece folder can fill the icon, VD and paperdolls together from a correctly named folder; chapter 14 gives the file convention.

Action coverage and gameplay fields

Open Animation preview and Load / refresh appearance. Select an action, direction and preview rate, then Show sequence or Play. The table shows which of the five stored directions are present. The shared cross marks the registration origin; the equipment layer is shown alone here.

Allow incomplete action set is an advanced option after you have reviewed missing sequences. It does not waive required walk/run/stand coverage or a weapon's required attack. For a release-quality wearable, inspect movement, riding, attack and both character/paperdoll variants as applicable.

Optional bonuses, penalties and on-hit spells

Before Review complete import, open Bonuses and effects. Select a preset, enter its Amount or Chance %, then choose Add / update effect. Selecting an existing row loads its values for editing. Adding the same preset updates it instead of stacking duplicate rows. Remove selected deletes one choice; Clear effects returns to ordinary equipment. The script preview shows the additions, and the final review shows them in the complete item definition.

Preset	Example and behaviour
Strength, Dexterity or Intelligence	-5 Strength lowers the wearer's strength while equipped; +5 increases it
Luck	+100 Luck while equipped; the shard's loot rules determine its benefit
Physical, Fire, Cold, Poison or Energy resistance	+10 increases that resistance while equipped; damage rules and caps belong to the server
Damage increase, Hit chance or Defence chance	Add an equipment combat modifier; the shard still controls the formula and caps
Skill bonus	Enter 5 for +5.0 effective skill or 2.5 for +2.5; up to five skills per item
On hit: Flamestrike, Fireball, Lightning, Harm, Magic Arrow, Energy Bolt or Explosion	Weapons only: choose 1-100% and spell power 1-1000

For example, choose On hit: Flamestrike, enter 15 in Chance %, and leave Spell power at 500. This adds a separate 15% roll to each successful weapon hit. Power is Sphere spell strength in tenths: 500 means 50.0. It is not guaranteed damage; the selected spell and shard rules determine the outcome. Each selected spell rolls independently, so more than one can activate on the same hit. The ordinary weapon damage continues.

These presets target Sphere X. Stats, Luck and resistances use its native equipment properties, which the engine adds and removes as equipment changes. Elemental resistance requires the server's elemental combat rules; Studio does not change server configuration. Skill bonuses use temporary effective-skill modifiers with matching unequip and item-removal cleanup. They preserve trained skill values and stack with other equipped preset items. Avoid editing a generated skill helper while its item is worn; unequip it before updating scripts.

The choices belong to the individual piece. In Armor set, Edit selected piece opens its saved effects. Copy selected piece's effects to all pieces replaces every piece's effect list with the selected list. This applies a bonus per piece, not once for wearing the complete set: copying +5 to eight pieces can total +40. Review that total before saving. Folder-imported pieces start with no optional effects.

Generated scripts remain editable in the normal Sphere scripts workspace after the reviewed import. The Test Scene and paperdoll display artwork; they do not execute these combat effects. Test equip, unequip, repeated swaps, skill stacking and spell balance on a separate server before distributing equipment.

On Server script, inspect the identifier, category and destination. Review complete import combines the new icon, TileData, imported appearance assets where applicable and script. It does not create missing artwork or automatically generate female animation substitutions.

Expand equipment capacity if needed

If no suitable primary equipment block exists, enable Expand equipment capacity if needed before finding space. Studio can append blank primary animation-index records for otherwise unused equipment IDs up to 2047. The expansion is shown in the review and occurs only when saving.

Existing references, mappings and occupied records stay protected. This option does not expand TileData, every animation bank or paperdoll allocations, and it does not author missing frames. You still need a free icon and both paperdoll destinations. Recovery of the reviewed operation restores the original index length when applicable.

14. Complete armor sets and whole-folder import

Open All tools > Complete armor set. A set groups independently wearable pieces into one reviewed save. Each piece still needs its own icon, animation appearance and paperdolls, or an explicitly reused compatible appearance.

Build a set manually

1. Enter Set name.
2. Choose a piece such as Helmet, Gorget, Chest, Arms, Gloves, Legs, Boots, Cloak or Robe.
3. Click Add piece. The equipment editor opens with an appropriate layer and armor category.
4. Fill in its icon, appearance and properties. Check animation coverage.
5. Choose Check and add piece to set to return to the set list.
6. Repeat for the remaining pieces.
7. Select any row to Edit selected piece or Remove selected piece.
8. Choose Review complete set and inspect all destinations and scripts before saving.

Adding or removing a row only changes the pending set form. It does not write or delete game files. Suggestions exclude pieces already in the set. Duplicate destinations, layers, animation IDs, identifiers and filenames are checked before accepting the combined import. Piece scripts normally use `scripts/armor`, Category Armor and the set name as Subsection.

Load a whole set at once

Choose Open set / collection folder and select an extracted folder, not its ZIP. Select the set from the list, then Add whole set. Studio loads matching files and proposes fresh destinations for all pieces. Edit the resulting rows and Review complete set.

A collection can describe multiple sets in manifest.json. Select the intended set; Add whole set does not mean install every collection entry. If a layer is already present in the current form, remove or edit that piece before adding another set with the same layer.

Enable Expand equipment capacity if needed in the set builder before loading pieces when the existing primary equipment allocation is too small. The option is carried into the piece requests and remains subject to the checks described in chapter 13.

Simple folder convention

A manifest is optional when the folder contains recognised named piece subfolders. Each piece folder needs exactly one of each of these assets; PNG image names may use BMP equivalents:

```
ExampleArmor/
  head/
    equipment.vd
    inventory-icon.png
    paperdoll-male.png
    paperdoll-female.png
  torso/
    equipment.vd
    inventory-icon.png
    paperdoll-male.png
    paperdoll-female.png
```

```
  gloves/
    equipment.vd
    inventory-icon.png
    paperdoll-male.png
    paperdoll-female.png
```

Recognised folder names include head/helmet/hood, gorget/neck, torso/chest, arms, gloves, legs/pants, boots/shoes, cloak, robe, waistcloth/belt, skirt and shirt. An unknown piece folder requires a manifest with its suggested layer or manual addition.

Manifest structure for a collection

The manifest contains a sets list; each set has a name and pieces. Each piece names its suggested_layer and four relative file paths. For example, this describes one helmet using layer 6:

```
{
  "sets": [
    {
      "name": "Example Armor",
      "pieces": [
        {
          "name": "Helmet",
          "suggested_layer": 6,
          "files": {
            "inventory": "head/inventory-icon.png",
            "animation": "head/equipment.vd",
            "male_paperdoll": "head/paperdoll-male.png",
            "female_paperdoll": "head/paperdoll-female.png"
          }
        }
      ]
    }
  ]
}
```

Paths must stay inside the selected folder. Each manifest set supports 1 to 16 pieces and valid equipment layers. Do not put absolute personal paths in a distributed manifest. Original source/reference IDs are not automatically used as installation destinations.

Before distributing a set

Check each individual piece, then the intended combination: backpack icons, male/female paperdolls, world standing/walking/running, mounted poses and attacks where applicable. Confirm that the files represent separate pieces, not the same complete-suit VD copied repeatedly. Keep the source files and manifest so recipients can reallocate for their client.

15. The Animations viewer and exports

Open Animations to inspect saved body sequences. Enter a body ID in decimal or 0x hex, or use Selected creature to follow the selected CHARDEF. Choose the source, action and direction, then Load body.

Auto uses supported routing; an explicit bank helps diagnose which physical animation is being displayed. Source/status text explains available resolution information. A missing action can mean the chosen body/layout does not contain it, rather than a broken entire archive.

Playback controls

Play/Pause starts or stops the loaded sequence. Previous and Next step frames. The slider scrubs the sequence and the frame label identifies the current position. Loop repeats it.

Base FPS and Speed combine into the effective preview rate. Changing them affects Studio playback and the timing of a saved GIF, not the native movement speed of a creature in game. Inspect both a normal-speed loop and individual frames for alignment problems.

Export choices

Action	Result
Export frame	The current animation frame as an image
Save GIF...	The displayed sequence as an animated GIF with preview timing
Export body PNG / VD...	Opens bulk export with the current body selected
Import frames / GIF / .vd...	Opens the animation importer for reviewed MUL writes
Right-click > Show in Test Scene	Sends the loaded body, source, action and direction to the local scene

PNG/GIF exports are useful for image editing and demonstration. A complete VD export from a MUL body preserves native animation information more directly. Keep body.json/frames.json metadata with exported folders when you intend to reimport them.

Modern UOP animation sequences can be previewed and exported to PNG, but they do not have a universally interchangeable action layout with MUL. Studio does not guess a whole-body VD mapping for them. Display directions 5 through 7 can be mirrored views of stored directions, not additional independent sequences to write back.

16. Animation workshop and layered preview

Edit frames

Open All tools > Animation frames / body reimport. Load displayed sequence takes the current Animations view. Open exported sequence reads a frames.json-based sequence. Open frame images reads supported PNG/BMP frames or GIF.

Set destination bank, body, action and stored direction for an eventual MUL import. These fields identify the write destination; they do not themselves change the currently loaded source files.

Use the slider to choose a frame. Origin X/Y set the registration point. Apply origin to frame changes one frame; Apply origin to all frames changes the sequence. The ground-origin guides help identify unwanted shifts. Show neighbouring frames provides an onion-skin comparison.

Move frame earlier/later changes order. Crop transparent border trims the selected frame while maintaining the intended registration. Replace this frame substitutes its image. Undo reverses supported workshop edits. Play / Pause checks motion at Preview FPS.

Export edited sequence preserves an editable image/metadata folder. Do this before closing unfinished work; general session restoration does not include the workshop's unsaved frame canvas. Review import to MUL prepares the native sequence write. Allow replacing existing sequences must be enabled for an occupied destination.

Reimport a body folder

Export the body first using the bulk/body exporter. Edit the PNGs while keeping the folder structure and metadata. Reimport body folder reads the body.json, Action and Direction arrangement and preserves supported origins/index information.

For a new body, keep Body folder: use next free block enabled and choose a compatible bank. For an update, switch off automatic body choice, enter the intended destination and enable replacing existing sequences. Review the complete list before saving. A dropped body folder opens the workshop but still requires this destination setup and review.

Mirrored directions exported for viewing are not reimported as new stored directions. A body exported from modern UOP needs deliberate action mapping; this feature is not automatic complete UOP-to-MUL conversion. Use individual supported sequences with explicit destinations if necessary.

Equipment / mount preview

Open All tools > Equipment / mount preview to combine multiple saved animation layers. Add the mount first, then rider/body and equipment. Each layer has its own Body / equipment ID, source, action, direction, hue and X/Y offsets.

1. Enter a body and its source/action/direction, then Add layer.
2. Add the rider or other body using the appropriate pose.
3. Select a wearable in Item artwork and choose Add selected item, or enter its equipment animation ID manually.
4. Select rows to Remove layer, Move layer earlier, or Apply offsets / hue.
5. Play / Pause to inspect the combined motion.

The layers share a ground origin, but you control their order and poses. Add selected item reads the item's TileData animation link. It does not import a new wearable. Offsets, hues and layer order are preview settings and are not

written to native assets or Sphere scripts. This is useful for checking overlap, not a full implementation of the game renderer's equipment rules.

17. Gump artwork: browse, recolour, replace and import

The Gumps tab edits interface artwork records. These images can be backgrounds, buttons or paperdoll pieces. The Sphere gump designer in the next chapter arranges such artwork into a dialog; the two tools have different purposes.

Browse and use a gump

Choose Auto, MUL or UOP and Reload. Search gump ID/name and use All, Free candidates or Used. Scroll thumbnail cards and select one for the full preview. The preview offers Fit and fixed zoom levels.

Enter Preview hue ID and Apply, or Choose hue. Hue 0 shows original colours. Export PNG saves the displayed preview, including its selected hue. Use hue 0 if you want an uncoloured source export. Copy Sphere GUMPPIC line gives a starting command for script editing; position and surrounding dialog logic still need to suit your layout.

Right-click a gump preview/card for image replacement, references, favourite actions and export where offered. Inspect equipment-linked gump IDs before replacing them: a paperdoll slot may be referenced through an equipment animation even if it is not named in a simple dialog script.

Import a batch

Click Import PNG / BMP. Choose the images and destination format. Choose automatic consecutive IDs or deliberate IDs from filenames; review their order and proposed destinations. Browse slots / choose a free range helps find candidates.

Enable Replace the listed gump slots only for intended replacements. That option can include referenced IDs, so review their usage. Inspect the converted image and open the save review, then queue or save. This changes gump artwork; it does not automatically add a Sphere dialog that displays it.

Replace one existing image

Select the correct gump and use its right-click Replace ... with image action. Choose the PNG/BMP and compare the old and converted new image. Keeping the ID means existing scripts/paperdoll links can continue to refer to it, but changed dimensions or alignment can affect every place it is used. Preserve the full intended canvas for paperdoll pieces.

If a gump looks absent or unchanged after saving, verify the active source format, the client's actual data folder, the ID used by the script and any overriding files. Reload the preview after disk changes.

18. Sphere gump designer

Open All tools > Sphere gump designer to arrange a dialog visually and generate Sphere source. It uses existing gump graphics from your client. Import new graphics in Gumps first when required.

Create a dialog

Set Dialog DEFNAME to a unique name beginning `d_`, such as `d_example_menu`. Choose Canvas width/height and Preview page. Supported canvas dimensions are 100 to 4096; the design supports up to 1000 controls.

Choose a control type, fill its properties and click Add control. Click/drag a control on the canvas to move it. Apply properties updates the selected control after editing fields. Refresh preview redraws; Undo and Delete control manage layout edits.

Control	Main properties	Generated purpose
Image	x, y, graphic	GUMPPIC artwork
Tiled background	x, y, width, height, graphic	Repeated gump artwork over an area
Text	x, y, hue, text	A text label
Button	x, y, graphic, pressed, id	Reply button with normal/pressed graphics
Input	x, y, width, height, hue, id, text	A text-entry field

Page assigns a control to a dialog page. Preview page chooses what you inspect; page 0 is shared content. Text is single-line. Buttons require distinct nonzero reply IDs; inputs require distinct input IDs. A reply button is not automatically a next-page navigation button.

Save the editable design

Save design writes JSON for future editing. Open design loads it. Save this separately from the generated .scp; editing the generated script does not update the JSON layout. General session recovery is not a substitute for saving the designer canvas.

View, export or install its script

View script shows the generated DIALOG and button-handler sections. Export script writes that text to a chosen output file. That export does not register the file with your server.

Install script prepares a reviewed new .scp inside the active server's scripts directory and a resource entry when needed. Choose a unique filename and dialog DEFNAME; existing dialogs are not silently replaced. Queue or save the reviewed operation, then open the saved script for custom behaviour.

Generated button handlers are stubs: they include a comment for your Sphere action and a return. You must implement what each button does. The desktop preview also uses desktop font metrics and cannot verify the game's exact text layout, input behaviour or your server-side handlers. Test the dialog in the target client after scripting it.

19. Hues and colour testing

Open Hues to browse the client's colour tables. Reload refreshes them. Search by name/ID, choose a hue and compare its 32-swatch palette with the image preview.

Game hue 0 means no colour transformation. File palette index 0 corresponds to game hue 1. Use the game hue ID shown by Studio when copying a Sphere COLOR assignment.

Preview a saved hue

Test image shows a standard sample. Load PNG uses your image. Selected item and Selected gump use available assets from those views. Preview grey pixels only limits the comparison to grey pixels; it does not change the item's saved PartialHue flag.

Copy Sphere COLOR copies a script assignment. Export preview PNG saves the coloured preview. Export hue list CSV exports the hue listing, not a replacement hues.mul. Import palette / Export palette use the supported UOFiddler palette-text interchange.

Edit or copy a palette

Select a hue and Edit hue. Change its short name, select one of the 32 swatches and use Set RGB hex or Colour picker. Create gradient fills the palette between chosen dark/light colours. Import UOFiddler palette can populate it from a file.

The preview uses the unsaved working colours. Test this unsaved hue on clothing / items opens a sample comparison without saving it first. Review hue changes shows the proposed write to that hue record; then queue/save.

Copy to slot prepares another palette record from the current one. Choose the destination carefully: a blank table is only a candidate and may still be used by world objects or scripts. Copying a palette does not automatically update the COLOR of items that used the source hue.

Stored fields includes TableStart and TableEnd. These are stored 16-bit fields, not the indices of the first/last swatch you want to edit. Preserve them unless you have a specific reason to change or import them. The reviewed write updates the selected hue record, retaining other records and block information.

Compare clothing and items

Try clothing / items lets you select up to 24 items by searching Sphere names/categories or using Item IDs / range. Choose Item art, Male clothing or Female clothing. Choose Item flags, Full hue or Grey pixels only to control the comparison.

The clothing modes show individual paperdoll pieces, not a complete animated dressed character. Choose hue switches the palette. Export comparison PNG saves the resulting board. Check more than one material: a hue that reads well on a grey robe may be poor on an already coloured weapon.

20. Sound workshop and attachments

Open All tools > Sound workshop. Select Auto, MUL or UOP, then Reload. Search the table by ID or name. Selecting a sound shows its waveform and duration. Play and Stop audition it through Windows audio.

Export sounds

Export WAV saves the selected sound with a WAV header. Export all WAV writes the available sounds to a new output folder. A sound archive's internal bytes are not themselves a complete WAV file; use the exporter rather than renaming raw archive data.

Import or replace audio

The WAV import processing settings apply to the files you subsequently choose:

Setting	Meaning
Trim start	Start position in seconds
Trim end	End position in seconds; blank keeps the end of the source
Volume multiplier	1 keeps level; lower reduces volume; higher may clip
Fade in/out	Fade duration in milliseconds

Choose Add WAV files for new candidate IDs, or Replace selected for one existing sound. New allocation considers supported archive entries, mappings/overrides, literal sound references and pending reservations. A replacement affects existing uses of that sound ID.

The importer accepts uncompressed PCM WAV at supported mono/stereo 8/16/24/32-bit depths. It converts to the native signed 16-bit mono, 22050 Hz sound payload. MP3 and floating-point WAV need external conversion to supported PCM first. Resampling is simple; prepare demanding audio work in a dedicated editor.

Inspect the duration and clipped-sample count in review. Play converted sound auditions the prepared payload; Stop stops it. With a batch, the review's audio audition is the first prepared sound, so inspect individual sources/results when validating a collection. Queue/save only after checking destination IDs and conversion.

Shorten a source filename if the derived sound name exceeds 39 Windows-1252 bytes or contains unsupported characters. Renaming does not change the sound's gameplay assignment.

Attach a sound to an item, creature or mount

Use this workflow to connect audio to an existing definition. Finish or discard open script drafts first. Scripts in a server ZIP are read-only; extract and link the server before attaching sounds.

1. Select the item, wearable, creature or mount in the definitions list.
2. Open Workspace and choose Attach sound to this asset... in the Sound in UO panel. Right-click > Attach sound... on a definition, or All tools > Attach sound..., also opens this window.
3. Check Attach to. It identifies the exact script section and line; other supported sections in that script are available in the list.
4. Choose Play when using the choices below.

5. Choose Auto, MUL or UOP for the saved sound source. Search by name, decimal ID or `0x` hexadecimal ID, then select a sound.
6. Press Play new sound or double-click the sound to hear it. Stop ends playback. The waveform and duration help you inspect the selected audio.
7. Choose Review sound attachment.... Read the sound ID, target, event and before/after script text. Listen to the review's audio if needed.
8. Choose Queue changes to defer the reviewed operation, or Save to UO files to save it now. Enable Keep a backup if you want retained originals for later restoration.

Save to UO files writes to the client-data and server-script folders linked by the active profile, after the safety checks. You do not have to paste the sound fields into the script yourself. It does not upload to a website or reload a running server; restart or reload the intended client/server after saving so the sound is used in game. For a new WAV, players also need the updated sound archive in the client data they actually use.

Target	Play when	Script connection
Creature or mount	Idle	SOUNDIDLE
Creature or mount	Notice / alert	SOUNDNOTICE
Creature or mount	Attack / hit	SOUNDHIT
Creature or mount	Get hurt	SOUNDGETHIT
Creature or mount	Death	SOUNDDIE
Item or wearable	Use / double-click	@DClick
Item or wearable	Equip	@Equip
Item or wearable	Unequip	@UnEquip
Effect / function	When this function runs	Start of the selected FUNCTION

A saved sound is reused without rewriting its archive. The definition or function supplies the gameplay connection: attaching to an item does not make it equipable, and attaching to a mount does not add vehicle physics or engine timing. Sound 0 is not offered because it represents no sound in Sphere.

You can also select a sound in Sound workshop and choose Attach selected sound to a thing..., or right-click the sound and choose Attach sound.... Select the intended definition, or open its script, before using that route. Check Attach to again before review.

See and replace sounds already associated with an asset

Open Attach sound and select Current sounds. The list shows the event, saved sound ID and name when available, and the script location supplying the link. Creature rows include sound fields inherited through a resolvable ID and the action sounds derived from a base SOUND set. Expressions and special server-chosen sounds remain labelled instead of being guessed.

1. Select the current sound row. Play current sound auditions that saved audio without changing the replacement you have selected.

2. Click Choose replacement. The Change control selects Set / replace selected sound and the Choose new sound tab opens.
3. Select another saved sound, or choose a ready-made WAV. Play new sound lets you compare the replacement with the current sound.
4. Choose Review sound attachment. Check the old-to-new value and the exact script change.
5. Choose Save to UO files or Queue changes.

Replacing an item or function sound updates only the selected command, keeping its object prefix, conditions, other arguments, comments and position. Other sound commands remain. Replacing an inherited creature action creates a local override for that creature; the parent definition stays unchanged. If the local field appears before ID, Studio moves the override after that inheritance copy so it takes effect.

Replacing a link does not overwrite the old shared audio slot. A WAV receives a free slot and is saved together with the changed link. Other assets that use the old sound keep it. To deliberately change shared audio everywhere, use Sound workshop > Replace selected instead and review all affected uses.

For an item or function, Add a sound preserves the existing commands and inserts or updates Studio's marked attachment at the selected event's start. Use Set / replace selected sound when you want to swap an existing command. External event scripts, called functions, runtime property changes and dynamic expressions are not fully evaluated here. An unresolved entry does not prove that the asset is silent; inspect its script or verify the event in game. Client sound.def or loose-file overrides can also affect playback.

Attach a ready-made WAV in one save

In Attach sound, choose Choose WAV file... instead of selecting a saved sound. Select an uncompressed PCM WAV, then use Play new sound to hear the converted audio. The supported input formats and filename limits are the same as the sound importer above. The attachment window converts the file without offering the workshop's trim, volume and fade controls; prepare those edits beforehand if needed.

Studio chooses an available, unreserved sound slot for the WAV. Review shows that new ID and the script connection together. The sound data and script edit are saved or queued as one recoverable operation. Existing sounds and pending reservations are considered when choosing the destination. If no suitable slot is available, reuse a saved sound or choose another compatible archive format.

Selecting a saved sound after choosing a WAV switches the attachment back to reusing that saved sound. Check the source note and review to confirm which audio will be used.

Attach a sound to an effect function

1. Open the effect's .scp from the Script directory so the whole file is available.
2. Choose All tools > Attach sound....
3. Under Attach to, select the intended Effect / function entry. A script can contain several functions; use the identifier and line to choose the correct one.
4. Choose When this function runs, select a saved sound or WAV, and review the attachment.
5. Save or queue it, then verify the event with the target client and Sphere after their data is reloaded.

The inserted sound runs at the start of that function and uses the object calling it as its source. A function invoked without a world object may need a manually chosen sound source in its script. An artwork ID or animation frame alone does not execute audio; merely placing an effect item does not call a separate effect function.

Test Scene remains a visual preview. It does not execute Sphere events or automatically play attached sounds. Use Play new sound in the attachment window to audition the audio; verify event timing in the target game setup.

Existing scripts, inheritance and repeat attachments

For a creature, Studio sets only the selected sound field on that definition. Other fields, inherited links and triggers remain intact. A local sound value overrides the corresponding inherited value without rewriting the parent definition.

With Add a sound for an item event or function, Studio adds a marked SOUND command before existing conditions and returns. Existing custom logic and other sound commands remain. This can play audio even if a later condition rejects the action, or produce more than one sound if the script already plays audio. If the sound should run only after a successful action, deliberately move the command to that point in Sphere scripts and review the logic. Set / replace selected sound keeps the chosen command at its current position.

Using Add a sound again on the same event updates Studio's unchanged marked command instead of adding another copy. That insertion mode requires manual review of duplicate triggers or edited/moved markers. To replace an existing command, select its exact row in Current sounds; Studio does not choose a custom command for you.

If the script already has a queued edit, save that edit or remove it from the queue, then reopen Attach sound. A saved sound with a queued replacement also needs to be resolved first. On-disk changes invalidate the attachment review; reload the current files rather than overwriting newer work. To add several sounds to the same script, finish each save before reopening the next attachment.

The browser previews the selected saved MUL/UOP archive. A game client's sound.def mapping or loose-file override can change what actually plays. Music is outside the sound workshop; verify the intended event, source and volume in the target client/server.

Copy creature sound fields manually

Creature sound fields opens SOUNDIDLE, SOUNDNOTICE, SOUNDHIT, SOUNDGETHIT and SOUNDDIE entries. Enter the intended sound IDs and Copy Sphere fields. Paste the assignments into the appropriate CHARDEF and save the script separately.

This helper remains available for manual script work. For a reviewed direct attachment, use Attach sound... above. Importing audio alone does not assign it to every creature automatically.

21. Test Scene: an in-app visual sandbox

Open the Test Scene tab, All tools > Test Scene (in-app), or Preview in Test Scene beside the definition list. Right-click item art, an Asset spaces card or the loaded animation preview for Show in Test Scene.

The scene runs entirely inside Studio. It can show characters, mounts, items, equipment and effects using saved files, supported queued edits or an unimported preview file. Its movement and obstacles are local; Sphere combat, ownership, taming, feeding and script triggers are not executed.

Attached sounds are not played automatically in this scene. Audition them in Attach sound or Sound workshop; check their actual gameplay event using the target client and server.

Load content

Load selected definition follows the current selection. Alternatively choose Kind, enter its ID and Source, then Load / refresh preview. Character/Mount IDs are animation bodies; Item/Equipment/Effect use their relevant item-art links. Read the source note above the scene to see what was loaded.

Include queued changes overlays supported reviewed animation, item-art, effect and equipment metadata/paperdoll changes in memory. Enter the proposed destination ID when testing a queued import. Refresh after changing the queue. This preview does not commit the queue to disk.

Preview VD / PNG / GIF opens supported source files before importing. Return to saved / queued files stops using that external preview file. A source-file preview cannot invent missing directions or actions.

Character and mount controls

Choose Idle, Walk, Run or Custom pose. Custom uses the Action ID. Direction selects one of eight displayed directions. Player body selects the reference/rider body where that control applies. Hue changes the preview colour; FPS and Speed change playback.

Mount mode offers Mounted rider and Rider X/Y offsets. Show reference player adds an optional comparison player in other suitable modes. Equipment mode uses the item's equipped animation and retains the player needed to display it. Missing required player art may be represented by a labelled placeholder.

Play/Pause controls animation. The frame buttons pause and step. Click the floor or focus the canvas and use arrow keys to move. Space toggles playback. These controls affect the preview, not native frame data or server speed.

Place an item without replacing the character

Selecting an item for Show in Test Scene prepares ground placement while retaining the character/mount already in the scene. In Room, choose Place item and click floor tiles to place copies. Use Move character when you want clicks to move the actor again. Escape returns to movement.

Wear in Test Scene, available on the small paperdoll, is the clearer route for an equippable item: it shows the item on the human model with the corresponding paperdoll. Ground placement and wearing are different actions.

Effects

Effect frames 0 uses AnimData when available. A positive count previews explicit consecutive art frames. Effect style selects Ground, On player or a looping Projectile demonstration. Check hue, scale, frame order and overlap. This does not verify the generated Sphere effect function's damage or timing rules.

Room and wall controls

Control	Use
Scene tool	Move character, Place item, Add wall or Remove placed
Wall direction	Choose Left or Right orientation for added walls
Lighting	Day, Dusk or Night preview brightness
Room	Client stone, Plain grid or Custom environment
Walls / local collision	Toggle the room walls/local obstruction behaviour
Floor grid	Show the floor tile grid
Place selected item	Prepare the selected item's art as a prop
Selected item as floor / wall	Use selected art for that environment role
Reset room	Clear the local arrangement and reset the room

The default walls follow the two back edges. Add wall lets you place test obstacles by clicking tiles; Remove placed clears placed items/test walls at the clicked tile. Environment art uses available client pieces, with a plain-grid fallback. Floor art is fitted to the scene's ground diamonds.

Zoom, snapshots and persistence

Auto zoom fits smaller windows and limits enlargement; Fit fills the available space. Fixed zoom choices provide closer inspection. Snapshot exports the rendered image as PNG.

The scene arrangement, movement, rider offsets and lighting do not change scripts, game files or queued imports. A snapshot is a demonstration image, not a saved playable world or editable house. Use House Builder when you want a design that can be compiled into a house/deed.

22. House Builder and housing deeds

Open All tools > House Builder + deed. This is a separate in-app editor for a fixed house assembled from existing item artwork. It saves editable .house.json designs and can compile a native multi with a matching Sphere house and placement deed.

Start a design

In House settings, set House name, Plot width/depth, Storage limit and Deed value. Apply plot settings applies validated changes. Plot sizes range from 3 by 3 to 31 by 31; piece heights range from -20 to 100. A design can contain up to 10000 pieces.

Choose Building pieces to search native art by name or ID and filter by role. The material selector offers matching groups such as timber, stone, brick or logs with associated roof/floor/door pieces, plus All building pieces, Favourites and Recent. Available results depend on the client's artwork. The star button toggles the chosen art in favourites.

Pick the correct role

Artwork ID selects the image; Role tells the builder what that piece does in the design. Inferred roles are a starting point, so check them. Floor, wall, roof, stairs and decoration are static pieces. Door and sign are functional components for the compiled house; the selected native art must support that function.

Every compiled design needs floor content, at least one working door and exactly one house sign. The interface suggests a standard brass sign, but the selected server must actually have a compatible inherited sign definition. A sign-like image assigned Decoration is not a house sign.

Drawing and editing tools

Tool/control	Action
Paint	Click or drag to add the chosen art at the active height
Line	Drag between endpoints to draw a line of pieces
Rectangle	Drag to fill a rectangular area
Perimeter	Drag to draw the rectangle's boundary
Erase	Remove the hit piece; right-click also erases a hit piece
Select	Click a piece, or drag a selection area at the active height
Eyedropper	Pick a piece's art, role and height
Copy / Paste	Copy selected pieces, then place their relative arrangement
Delete	Remove selected pieces from the design
X-/X+/Y-/Y+	Move selected pieces one tile in the chosen axis
Lower / Raise	Move selected pieces one height unit
Undo / Redo	Step through design changes

The translucent ghost shows prospective placement, including copied groups. Cursor text gives tile X/Y and height. An out-of-bounds paste preview is marked differently; the design still has to pass validation. Large previews may show only an initial portion of the placement group, so review the finished selection too.

Floors, visibility and selected-piece properties

Height chooses the active building plane. Presets include Ground 0, Main floor 7, First floor 27, Second floor 47 and Roof 67. The -20/+20 buttons move the plane quickly. These are convenient heights, not an automatic multi-storey generator.

Roofs toggles roof visibility. Upper floors controls whether pieces above the active height are shown. Use Fit or a fixed zoom, middle-drag to pan, and Centre view to reset pan and fit the plot.

Open Selection after selecting pieces. Artwork ID, X, Y, Height and Role describe the selection. For a group, coordinates reposition the first piece while preserving the other pieces' relative arrangement. Blank artwork or role preserves each piece's value. Apply to selected pieces validates the result and records an undoable change.

Save, open and recover editable work

Save project writes .house.json; subsequent saves reuse its path. House settings > Save project as creates another editable file. New and Open project handle unsaved work before replacing the current design. Import blueprint reads supported unhued CentrED CSV into the current project; check inferred roles afterward. Export blueprint writes the supported tile data as CSV.

CSV is interchange data, not a fully registered Sphere house. It does not carry the project's ownership link or every editor setting. Per-piece hues and unsupported visibility flags are rejected for fixed-house import. Share a .house.json when you want to preserve the editable roles and settings.

House Builder writes automatic recovery copies while you work. House settings > Recover a project opens those copies; then Save project chooses a permanent filename. Recovery is a convenience, not a reason to stop keeping deliberate project files.

Save/Queue in the compile review leaves the builder open. Save the editable project afterward to retain its compiled-house link. Undo and redo preserve that destination link; they do not undo an already completed game-file save. Use reviewed-save recovery for a file rollback.

Check and walk through the house

Check building reports low headroom, unreachable floors/entrances and enclosed floor gaps. Problem pieces are selected where applicable. A compile also presents architectural warnings for review. Some openings may be intentional; read the warnings in context.

Start walk test places a reference character outside the plot. Use arrows/WASD or click a destination for a local route. E or Space opens/closes a nearby door. Step out of its tile before closing it. Open doors appear faded in the preview. Stop walking cancels movement.

The walk test uses native heights, stairs, doors, headroom and blocked-corner checks within a bounded flat-ground simulation. It is not a proof of actual terrain placement or Sphere permissions. Test roof visibility and different floor levels to see what the character is doing.

Compile house + deed

1. Save the editable project and verify its roles, entrances, stairs and sign.
2. Choose Compile house + deed and review any architectural warnings.
3. Read the chosen free multi ID, affected client files and complete Sphere script.
4. Confirm the house name, deed, storage/value and resource registration.
5. Queue or save. When both native multi representations are present, the compiler updates MUL and UOP together.
6. Save the editable project again to retain its link to the compiled house.
7. Load the changes on a compatible test shard, create the generated deed and test placement, entry, ownership and redeeding.

Scripts go into scripts/housing/custom. The selected server must already provide compatible housing systems. Compiling is not a complete replacement for Sphere's housing code, and the editor does not place the house in the live world for you.

Update a compiled house

Open the saved linked .house.json, make changes and compile again. Studio retains the same multi ID, script filename and named house/deed identity, even when the displayed house name changes. The review shows previous and updated script text.

The saved link verifies the destination script and native multi payloads. If they changed externally, belong to another project or no longer match, the update stops. Save queued changes before compiling another update to the same house. Keep the matching project with the target that owns that allocation.

For a house compiled in 0.23.0, open its original unchanged design and use House settings > Link an earlier compiled house. Studio requires an exact script/art match before reviewing the link. Save that link and the editable

project before editing further. A changed old house cannot be adopted merely by giving a new design the same name.

Already placed houses retain saved fixtures and region information. Redeed and place a fresh house to apply structural changes. Back up and plan deployment with the shard operator when houses have owners or stored belongings. Studio does not automatically migrate occupied world houses.

Share houses

Share the editable design and a preview image, plus required nonstandard tile assets and setup notes. A linked project targets its recorded compiled house; it is not a universal allocation for another shard. Preserve your linked original. For a fresh destination, create a new unlinked design, exchange the supported blueprint and recheck roles/settings, then compile against that destination's current data.

Generic .uoasset content-pack remapping does not support compiled houses in this version. House Builder is for fixed house designs, not full map editing, a networked CentrED server or the complete custom-housing protocol.

23. Find references and investigate shared content

Asset workspace > Find references and applicable asset right-click menus open a report for the selected asset. Script workspace also offers symbol-oriented reference navigation. Use these tools before changing a shared appearance or deleting old content.

The report lists file, line/record and evidence. Double-click a resolvable script/definition row to jump to it. Export reference report writes the report with its coverage description so you can retain or share the findings after removing private details.

Definition's reference list focuses on supported indexed relationships. The broader report scans script and mapping mentions; item scans also inspect supported unambiguous classic multi layouts and statics. Large scans run in the background and can be cancelled between units of work.

Interpret results carefully

The same number can mean an item, body, coordinate or unrelated value. Read the line instead of deleting every numeric match. A dynamic Sphere expression may use an asset without a simple literal ID. World-save objects, UOP multis, every override and complete runtime behaviour are not universally resolved.

A clean report is therefore not proof that an ID is unused. Combine it with allocation evidence, source inspection and knowledge of the shard's world data. If several definitions intentionally share an animation, choose a new destination for a visual change that should affect only one of them.

24. Remove definitions, animations or whole scripts

Select a definition and use Remove related content, or its right-click equivalent. Choose the removal scope before preparing the final review.

Scope	Intended result
Selected definition only	Remove that definition while keeping animations, art and companions; outside references may remain
Definition and related content	Consider its linked mount/figurine definitions and removable unused assets while protecting shared content
Entire animation group	Include other definitions using the same animation and their linked removable content

Review a removal

1. Choose the intended scope.
2. Read REMOVE and KEEP / REVIEW MANUALLY in the review.
3. Check every listed definition, art/animation slot, mapping and script-file change.
4. Resolve reported external references before requesting a larger removal again.
5. Choose Remove listed content only when the list matches your intent.
6. Refresh the directory/definition view and inspect the result. Keep the automatically retained backup until you have verified it.

Removal is deliberately reviewed and keeps a backup; it is not an unreviewed one-click purge. A larger animation-group scope is useful for genuinely shared custom content, but does not authorise deletion of arbitrary unrelated scripts that mention it.

Why a .scp may remain

A file may contain other definitions, functions or meaningful content. Removing one section does not necessarily delete the whole file. Cleanup can remove a now-empty script and related exact resource references when applicable, but shared content remains.

Open the remaining file and inspect it. Use Remove empty .scp file for genuinely empty content, or Delete script for an intentional reviewed whole-file deletion. Deleting a whole script keeps client assets and other scripts; it can leave references that you must update. The two operations serve different purposes.

What is not automatically removed

Dynamic references and existing world objects are not comprehensively cleaned up. Sounds, gumps and additional effect frames are not all inferred and removed automatically. Shared/ambiguous item uses or static placements may cause assets to be kept. Read the review's actual list rather than assuming “related” means every conceivable dependency.

If Studio reports an outside reference, open the named file/line. Decide whether to update the reference, retain shared content, or choose definition-only removal. Do not erase a whole decoration/worldgen file just to silence a reference warning. Use a retained backup to restore a mistaken completed removal as described in chapter 4.

25. Bulk export and resuming exports

The top-bar Bulk export tool exports saved item art and animation bodies. Animations > Export body PNG / VD opens it with the current body range selected. Pending edits are not included; save them first if they must be part of the export.

Choose export content

Option	Result
Item artwork as PNG	Exports static item images from the selected source
Item source Auto / MUL / UOP / Both	Controls which item representations are exported
Animation bodies	Exports bodies within the specified range
Animation source	All available, all MUL banks, a particular bank, or UOP
Individual PNG frames	Image files for supported action/direction frames
One .vd per MUL body	A complete compatible native body export
Include mirrored directions 5-7	Extra display-direction PNGs, not new stored sequences
Align PNG frames on a shared canvas	Consistent canvas within each action/direction

Set the body range using decimal or 0x values. The range can include bodies without Sphere definitions; empty slots are skipped. A full export may contain hundreds of thousands of files, so begin with a small relevant range when checking settings.

Run and inspect an export

Choose an existing output parent folder and click Export. Studio creates a new organised UO-Export folder. Item images are arranged by source/category/subsection and ID/name. Animations are arranged by source, body type, body, action and direction. Metadata records origins and source IDs beside the images.

Open export folder reveals the result. Check the completion status and error report; an export containing some successful files may still have skipped/unreadable assets. Original archives are not modified.

Keep body.json and frames.json information with their associated PNGs for editing/reimport. A VD preserves original compatible MUL data; modern UOP exports are PNG-based and do not silently receive guessed VD action mappings.

Cancel and resume

Cancel export requests cancellation between units of work. Retain the incomplete folder if you intend to resume it. All tools > Resume bulk export asks for the incomplete supported export folder and verifies its metadata and completed outputs.

Resumption reuses completed images only after verification. Changed source archives or manually edited exported images can stop resumption. Keep a separate copy for artwork editing rather than editing the folder you still plan to resume. If the inputs have intentionally changed, start a fresh export.

Bulk export is for saved artwork/animation images, not a dependency-complete server content package. Use Content packs for explicitly queued imports and their linked scripts, and House Builder project/blueprint files for editable houses.

26. MUL / UOP conversion and file checks

Convert supported UOP archives

Open MUL / UOP... from the main toolbar and choose Direction: UOP to MUL, or open All tools > UOP to MUL.... Add UOP files lets you select archives; Add from linked client finds supported archives in the active client's data. Select an unwanted row and Remove selected to leave it out.

Choose an output parent folder, then Convert. Studio creates a new UO-MUL folder with the converted files and a conversion report. Open output folder shows the result. Original UOP files stay unchanged; output is not automatically installed over the active client.

Supported conversion families include item art, gumps, sounds, maps and multis. Multi conversion uses the High Seas format. The original application's modern animation UOP viewer is separate: those animation archives are not converted by this utility.

Check the report and use the matching generated MUL/index pairs. Do not copy only a new index over unrelated data, or assume conversion updates the client's source preference. If both MUL and UOP are present, confirm which one the game and each Studio tool are reading.

Cancel conversion stops between supported work units. Inspect status before using any output; a partially populated output folder is not a completed conversion. Conversion is not map editing, a patching service or automatic client-version migration.

Convert MUL files to UOP

1. Open All tools > MUL to UOP..., or choose MUL to UOP in the main converter's Direction list.
2. Choose Add MUL files... and select the supported data files. Keep each matching index in the same folder; you select the data file, not its index. Add from linked client can find the supported data files for you.
3. Read the listed output filenames and remove any unwanted entries. A batch cannot contain two data files that would produce the same archive.
4. For multi.mul, choose High Seas (16-byte) under MUL multi format only when that is the source's verified format. Keep its original housing.bin and multi-components.txt beside it, as explained below.
5. Browse to an output parent folder separate from the working game installation, then choose Convert.
6. Wait for packing and verification. Studio reads every completed archive entry back and checks its payload before exposing the completed folder.

7. Choose Open output folder. The new UO-UOP folder contains the archives and conversion-report.json; read the report before using the results.

Selected data file	Required companion	Generated archive
art.mul	artidx.mul	artLegacyMUL.uop
gumpart.mul	gumpidx.mul	gumpartLegacyMUL.uop
sound.mul	soundidx.mul	soundLegacyMUL.uop
mapN.mul / mapNx.mul	None	mapNLegacyMUL.uop / mapNxLegacyMUL.uop
multi.mul	multi.idx, housing.bin, multi-components.txt	MultiCollection.uop

Occupied asset IDs and payloads are preserved. Empty index slots are not stored as assets in UOP, and archive compression, offsets and framing can differ from an earlier archive. Art is stored without compression; the other supported families use zlib. The output is verified as a readable archive, not as a byte-identical recreation of an original client's packer.

Only terrain is included when packing maps. Map statics, map differences, scripts, TileData, hues and other separate metadata remain separate. anim.mul and modern AnimationFrame/AnimationSequence conversions are not supported here. Changing the container does not create missing artwork, expand the client's capabilities or establish compatibility with every client build.

Preserve multi housing data

MultiCollection UOP needs information that is not contained in multi.mul alone. Extract the matching original MultiCollection.uop with UOP to MUL first and keep housing.bin and multi-components.txt together with the resulting High Seas multi.mul and multi.idx. The housing file contains the housing catalogue; the sidecar preserves component IDs used by structures and boats. Keep even an empty, header-only sidecar produced by the extractor.

The reverse converter requires both companion files and an explicit High Seas (16-byte) format choice. It rejects unsupported flag bits, mismatched sidecar artwork/positions, duplicate sidecar entries or references to missing tiles instead of discarding them. Older 12-byte multi.mul data is not supported. Do not choose High Seas simply to bypass the format check or invent empty companion files for an unknown source.

Conversion safety and using the result

Both directions check access to the required source files and output folder before creating temporary output. Sources must remain stable throughout conversion; another file-access check and source-revision comparison run before publishing the completed folder. Missing indexes, busy files, damaged records, changed sources or cancellation stop the batch without installing a partial result. Close programs using the reported files and retry when they are stable.

The converter always creates a new result directory. It does not overwrite the selected MULs/UOPs, switch the active profile to the output or install it into a live client. Changing Direction clears the selection and completed-result link; choose the new direction's inputs before converting again. Retain the original complete client data and test the result in an independent copy. Keep a local conversion report private if it includes personal source paths.

Check game files

Item artwork > Check game files is a read-only structural diagnostic. It inspects supported client file arrangements and reports issues such as recognised layout, inconsistent files and whether Sphere's MulFiles setting points to a different location than Studio's client folder.

Save the report when troubleshooting. It does not save pending edits or repair files. Include the exact Studio version, relevant action and the latest associated client/server error when asking for support; remove personal paths or private content from public reports.

If the check finds a mismatch, correct the profile or intended server configuration deliberately, then rerun it. Do not combine indexes/data from different installations or apply wholesale archived files merely because one program starts successfully afterward.

27. Compare projects and create independent test copies

Compare saved files or logical assets

Open All tools > Compare project files. Choose the other folder and the comparison type. Compare with linked client compares against the active client's saved data; Compare with server scripts performs a file comparison for the server source.

Files compares file-level differences and is read-only. Items, Gumps, Sounds and TileData compare supported logical records, so moved archive offsets alone do not necessarily count as changed content. This helps identify actual asset changes after an archive has been rebuilt.

Select a populated record in a supported asset comparison and use Review selected assets from other folder to prepare its import into the active client. Read the review carefully: the other folder is the source; the active profile is the destination. It does not replace a whole archive as a general synchronisation operation.

Export comparison report retains the results. File comparisons do not provide a universal merge tool for scripts, maps or entire client installations. Use the script editor for deliberate source changes and the appropriate importer for supported assets.

Make a test copy

1. Finish pending work and ensure the source files are stable; close programs that keep changing them.
2. Choose All tools > Create test copy.
3. Choose a parent folder outside the existing client and server folders.
4. Enter a new simple folder/profile name.
5. Wait for the verified copy and profile registration.
6. Select that new Shard profile when ready.
7. Read TEST_COPY.txt before launching anything from the copy.

The tool copies server/client data, excluding certain application backup/cache folders and linked-path cases, and checks for changing sources. It updates recognised MulFiles and ScpFiles entries in copied INI files to the copy. It does not rewrite every custom absolute save/log/database path or configure all network settings.

The optional client executable path is still shared as a remembered executable location. Verify its data-directory setup separately. Do not assume launching it automatically uses the copied assets.

A copied server can contain accounts, saves and configuration from the source. Keep the test copy private; it is not a cleaned public distribution or a provisioned virtual server. Review custom external paths before starting Sphere so testing does not write outside the intended copy. Studio does not start the copied server for you.

28. Portable content packs and distribution

Open All tools > Content packs to move explicitly reviewed content between projects. A .uoasset pack contains the pending bundle you chose to prepare, including supported linked scripts/mappings. It is not an automatic export of every dependency used by a selected definition.

Export a content pack

1. Prepare the required imports/edits and Queue changes instead of immediately saving them.
2. Inspect Pending changes and remove unrelated operations from this bundle.
3. Include the required icons, animations, gumps, hues, sounds and script changes deliberately.
4. Open Content packs and choose Export pending bundle.
5. Save the .uoasset outside your working client/server files.
6. Test importing it against a separate suitable project before distribution.

Exporting the pack does not save the queued changes to your game files. If you already saved and cleared the queue, the pack exporter cannot reconstruct an entire bundle just from the current definition. Prepare a deliberate supported export/import bundle again or distribute source assets with clear instructions.

Import a pack

Open content pack loads its inventory. Double-click an asset row to set an explicit destination ID in decimal or 0x hex. Review every remap, especially connected equipment IDs and consecutive effect frames. Enable Allow replacement of occupied asset IDs only for an intended replacement.

Review pack import shows destination mappings and linked scripts. Resolve conflicts, then queue/save using the normal workflow. Existing script filenames/definitions are not silently overwritten just because binary replacement was enabled.

Supported remapping and limits

Supported mappings include compatible MUL animation bodies, item/TileData links, known numeric Sphere fields and equipment paperdoll relationships. Animated-item frames must retain their relative consecutive arrangement.

The pack importer does not perform implicit MUL/UOP conversion, arbitrary TileData-layout conversion or universal rewriting of dynamic expressions, macros, custom functions and complex dialog code. Changed body IDs with body-mapping files may require the normal importer. Destination capacities and dependencies still have to match supported formats.

Compiled houses are not supported by generic pack remapping in this version. Use the editable House Builder workflow described in chapter 22. A pack also does not include all saved-world objects or prove that every runtime dependency was captured.

Prepare a recipient-friendly release

Distribute only the intended content and documents. Include a short start-here file explaining the tested Studio version, required client/server formats, dependencies, how to import, what to inspect and how to restore the reviewed operation if needed. Provide previews and original editable source files when you have the rights to distribute them.

For armor, a clean set folder/manifest lets recipients allocate fresh destinations. For animations, keep exported registration metadata with the frames. For houses, include the unlinked design/interchange instructions and any required custom tile art, rather than assuming your multi ID is free elsewhere.

Do not distribute project.json, sessions, account files, world saves, private logs, personal absolute paths or an entire private test-copy folder. Review generated scripts/manifests for dependencies and location assumptions. Include applicable licences for anything you redistribute. Studio's installed licenses folder describes its bundled software dependencies; it does not grant rights to unrelated content.

A release acceptance checklist

- The recipient can follow the instructions without your folder layout or accounts.
- The pack opens or source folder loads on a separate compatible project.
- Destination IDs are reviewed against that project's current data.
- Required scripts and resource entries are included and correctly loaded.
- Both inventory/paperdoll and world appearance are checked where applicable.
- Movement, riding and relevant actions are present for worn or rideable content.
- The retained-backup/recovery route is understood before production installation.
- Version, formats, known limitations and content licences are stated plainly.

29. Troubleshooting and common questions

I cannot see all the panels or buttons

Maximise or resize the window and inspect its scrollbars. Drag the main pane dividers so the script directory and central editor each have space. House settings has a scrollable area; Test Scene has a sidebar with Preview and Room tabs. The current release keeps save-review actions separate from its scrollable details. Check the title-bar version if your installed app still shows an older layout.

I keep having to uninstall before upgrading

Use the current Setup directly. It should detect the earlier installation and offer an upgrade/repair in the same location. Save and close Studio first. An older Program Files installation may need Windows elevation using the same account. Check that a shortcut is not launching another old copy elsewhere.

No free block fits, even though I see empty slots

The import may need a consecutive range or a whole compatible body block. Individual free cells scattered through an index do not form such a block. Check source/bank, layout, pending reservations and the batch size. Use another supported range or reduce an independent-image batch. New equipment can opt into its specific capacity expansion, but still requires icon and both paperdoll destinations.

Do not use Replace existing to get past an allocation error unless you actually intend to replace those records. A No UOP entry label is not permission to write unsupported UOP body animation.

The app says a file changed after review

The plan was prepared against an earlier revision. Another editor, a previous save or a queued/draft conflict may have changed its dependencies. Keep the current files, reload/review the intended operation and resolve older queue entries. A house update also needs its matching compiled-project link. Do not disable the check by copying stale archives over the target.

My asset looks correct in Studio but unchanged in game

Check the active profile's linked client folder, the game's actual data folder, and Sphere's MulFiles location. Check whether the client uses UOP while you edited MUL, or another override is taking precedence. Confirm the numeric art/body/gump ID used by the script. Save the queue, reload/restart the relevant program and test again. Use Check game files to gather evidence.

The character disappears, slides or faces the wrong way

Inspect the body bank, raw versus Sphere body ID, layout and action/direction. A single image sequence does not create all directions. Check ground origins and compare stand/walk/run. Display directions 5-7 may be mirrored. For a persistent correction, edit/export/import the frames; preview offsets do not modify the saved animation.

My mount is a vehicle, not an animal

Choose Mount > Server script > Mount behaviour > Rideable object when generating the new import. Use the generated inventory-item command. Double-click the item in your backpack to mount; dismount returns the item. Existing animal scripts are not automatically converted. Test ownership and mounting behaviour on the actual shard.

A wearable shows in my backpack but not on the model

An inventory icon is only one asset. Check TileData's layer/animation, the actual equipment VD, male/female paperdoll gumps and their source format. Load appearance and check action coverage. Use Wear in Test Scene for wearing; Show in Test Scene can prepare ground placement. Preserve full paperdoll canvases and provide each armor piece's own assets.

Can I split one complete armor VD into separate pieces?

The set builder does not split a whole-suit VD. Supply separately authored piece animations, individual inventory icons and male/female paperdoll pieces. If the source is intentionally one full-body costume, treat it as one wearable appearance rather than pretending it contains independent equipment layers.

The definition is still referenced outside the removal group

Open the reported script and line, read the use and decide whether it belongs in the removal. Entire animation group can include other definitions that share that animation, but unrelated external code still requires review. Update the reference or keep the shared content. Definition-only removal leaves assets/references for manual handling.

Removal did not delete the .scp file

Inspect the whole file. It may still contain unrelated sections or functions. Use Remove empty .scp file only when appropriate, or Delete script to review intentional whole-file deletion. That does not automatically delete the script's entire transitive asset set. Read REMOVE and KEEP instead of relying on the filename alone.

A script reports a 'charmap' decoding or encoding error

Keep the original bytes and identify the file named in the error. Reload with the current Studio version; supported script reads preserve normal UTF-8/Windows-1252 handling, but not every external file is guaranteed valid. For generated names, shorten them and remove unsupported characters where the client field requires Windows-1252. Do not resave an entire script collection with characters discarded. Include the exact error and a minimal non-private sample when seeking help.

A generated script says the filename or identifier already exists

Choose a unique named definition and filename across the script tree. Reusing an art base through ID inheritance is different from defining that numeric base again. For a linked House Builder project, use its update workflow; for an ordinary existing .scp, edit it deliberately in Sphere scripts. Do not create another conflicting copy simply by changing folders.

Save fails with access denied or a file-lock message

The Save safety check lists unavailable files before a reviewed save creates backups or prepares output. It can also detect another native archive in the selected client folder being used, even when that archive is not part of the edit. Close the client, Sphere and other editors using the reported files, confirm the profile points to a writable working location, then retry. Studio preserves Windows permissions; it does not reset them globally or close programs for you.

An initial failed check has made no partial game-file save. A second check can stop the operation after preparation but before replacement. If the message instead reports an interrupted save or requests recovery, keep its journal and follow Save / recovery centre before retrying. Chapter 4 explains the difference.

Attach sound asks me to save or discard edits

Finish open script drafts before opening the attachment window. If the same script has a queued change, save it or remove it, then reopen Attach sound against the current file. Finish a queued replacement of the selected saved sound first too. This avoids silently combining a new attachment with an older reviewed snapshot.

My attached sound does not play, or plays twice

Test Scene does not run sound events. First use Play in Attach sound to check the selected audio, then verify the actual item event, creature field or function in the target client/server after reloading. Check the selected script section and event, inherited overrides, sound.def mappings and loose audio overrides. An effect function needs to be called; an art ID alone is not a sound trigger.

An item event or function may already contain custom SOUND commands. Studio preserves those and adds its marked command at the start, so both can play. Review the script and deliberately adjust timing or duplicates. Do not delete unrelated event logic.

Sphere will not start after a save

Keep the latest Sphere startup error. Run Item artwork > Check game files and save its report. Compare the affected files and generated script/resource registration with the reviewed operation. Check client/server data paths and compatible formats. Restore the retained linked backup if rollback is appropriate and there have been no intervening changes. Do not repair an unknown problem by mixing unrelated archives and indexes.

My backup refuses to restore

A later edit may have changed one of the operation's files. Recovery stops rather than discarding that newer work. Keep both the backup and current files, identify the intervening changes and plan a merge or deliberate rollback. A whole-file recovery affects all changes inside that archive, not just one visible asset.

My house will not compile or update

Check the required floor, working door and exactly one compatible house sign. Check plot bounds, height, native artwork and matching server housing support. Review headroom/entrance warnings. For an update, use the saved linked project and save queued changes first. If linking a 0.23.0 house, start from its exact original design/name/settings before modifying it.

The updated house in the world still has old fixtures

The compiled definition can change while a placed world house retains its saved door/sign/region state. Redeed and place a fresh instance for structural changes, with appropriate handling of ownership and belongings. House Builder does not migrate occupied world objects automatically.

Does Test Scene prove my spell, pet or house works?

It proves only the aspects you actually inspect in that visual simulation. It does not execute Sphere events or fully reproduce terrain, collisions, equipment sorting or permissions. House Builder's local checks are also approximations. Use the target client/server for gameplay acceptance after visual preparation.

How do I ask for useful support?

Include the Studio version, tool/button used, asset type, selected source format and a clear reproduction sequence. Copy the exact error, state whether changes were queued or saved, and mention any other editors in use. Add the read-only file-check report where relevant. Remove credentials, accounts, world saves and personal paths before posting publicly.

30. Task finder and keyboard reference

Choose the right tool

I want to...	Start here
Find an item or creature definition	Left search and Category / subsection
Inspect linked assets and references	Workspace and Definition
Change a definition or whole script	Sphere scripts; Script directory for full files
Generate Sphere, RunUO, ServUO or ModernUO scripts for existing artwork	All tools > Generate server script
Edit several scripts or replace text	All tools > Sphere script workspace
Inspect an icon, TileData or paperdoll	Item artwork
Find candidate art/body space	Asset spaces or the importer's slot browser
Make a basic item using existing art	All tools > New Sphere item > Use existing artwork
Prepare an external Fiddler import kit	All tools > New Sphere item > Prepare new PNG for UOFiddler
Import static/animated item art	All tools > Static tiles from folder / ZIP
Import an effect and visual function	All tools > Animated item or effect
Import a monster body	All tools > Creature or monster
Make a mount or rideable object	All tools > Mount or rideable object
Import one wearable or weapon	All tools > Clothing or weapon
Import a complete armor set	All tools > Complete armor set
Play or export body animation	Animations
Edit frames or reimport a body folder	All tools > Animation frames / body reimport
Combine rider, mount and equipment	All tools > Equipment / mount preview
Replace/import interface images	Gumps
Design a Sphere dialog	All tools > Sphere gump designer
Edit/test colour tables	Hues

I want to...	Start here
Import, trim or export sound	All tools > Sound workshop
Link a saved sound or WAV to a thing	Asset workspace > Attach sound to this asset...
Play a sound when an effect function runs	Open its script, then All tools > Attach sound...
Preview assets in a room	Test Scene
Build a fixed house and placement deed	All tools > House Builder + deed
Generate a separate wilderness map	All tools > World Generator...
Remove a definition and related content	Definition selection > Remove related content
Delete a complete script	Script directory > right-click > Delete script
Export many saved images or VDs	Bulk export
Resume an interrupted export	All tools > Resume bulk export
Convert UOP archives to MUL/index files	MUL / UOP... > UOP to MUL
Pack MUL/index files into UOP archives	All tools > MUL to UOP...
Diagnose file/path mismatches	Item artwork > Check game files
Compare records between clients	All tools > Compare project files
Create a separate development copy	All tools > Create test copy
Share reviewed imports/scripts	All tools > Content packs
Restore drafts or a file backup	Asset workspace > Save / recovery centre

Keyboard and mouse actions

Shortcuts act on the focused control. Click the script editor or scene canvas first; typing in an entry field should not be confused with moving a character.

Context	Action	Result
Main window	Ctrl+P	Open Workspace
Sphere scripts	Ctrl+S	Validate/save the current script context
Script Find	Ctrl+F	Focus Find
Script Find	F3 / Shift+F3	Next / previous match

Context	Action	Result
Find entry	Enter / Shift+Enter	Next / previous match
Find entry	Escape	Return focus to script text
Script workspace	Ctrl+Space	Completion suggestions
Script workspace	F12	Go to definition
Multiple-selection lists	Ctrl/Shift-click	Select multiple rows where supported
Flag picker	Double-click / Space	Toggle selected flag
Item artwork ID	Enter	Inspect the entered art ID
Test Scene canvas	Arrow keys	Local movement
Test Scene canvas	Space	Play/pause
Test Scene canvas	Escape	Return to movement tool
House Builder canvas	Middle-drag	Pan view
House Builder canvas	Right-click	Erase hit piece
House Builder canvas	Delete	Delete selected pieces
House Builder canvas	Ctrl+Z / Ctrl+Y	Undo / redo design changes
House walk test	Arrows / WASD	Continuous local movement
House walk test	E / Space	Open/close a nearby door
House walk test	Escape	Stop walking

Generated script destinations

These defaults organise new files; they do not automatically move older scripts. Inspect the destination during review and keep filenames unique. The gump designer asks you to choose a new .scp inside scripts.

Import kind	Default script folder
Monster	scripts/monsters
Mount / rideable object	scripts/mounts
Item	scripts/items
Clothing	scripts/clothing
Weapon	scripts/weapons
Effect	scripts/effects
Armor set piece	scripts/armor
Compiled house and deed	scripts/housing/custom

31. Formats, glossary and version boundaries

Files you will encounter

File/family	Role in Studio
.scp	Sphere definitions, functions, dialogs and resource configuration
art.mul + artidx.mul	Indexed item artwork
ArtLegacyMUL.uop	UOP item artwork
tiledata.mul	Item/land metadata; Studio's editor focuses on supported item records
anim.mul/anim.idx and numbered banks	Supported creature/equipment animation sequences
AnimationFrame UOP family	Modern body-animation viewing/export sources
animdata.mul	Animated-static frame offsets/timing data
gumpart.mul + gumpidx.mul / gump UOP	Interface and paperdoll artwork
hues.mul	Colour-table records
sound.mul + soundidx.mul / sound UOP	Native sound effects
multi.mul + multi.idx / MultiCollection.uop	Compiled fixed structures/houses
.vd	Compatible UOFiddler animation-body transfer
PNG/BMP/GIF	Image and animation-frame source/export formats, according to the tool
PCM WAV	Sound import/export interchange
frames.json / body.json	Animation export registration and structure metadata
.uoworld.json	World Generator seed, settings and painted biome regions
mapN.mul / staticsN.mul / staidxN.mul	Matching native terrain and scenery set from World Generator
.house.json	Editable House Builder document, including an optional destination link
Gump design JSON	Editable visual dialog layout
Blueprint CSV	Supported unhued CentrED tile interchange
manifest.json	Set/collection description for supported armor-folder import

File/family	Role in Studio
.uoasset	Portable bundle of deliberately queued assets and linked content
project.json	Local profiles/settings; not public content

Keep data and index files as matching pairs. Preserve metadata next to exported frame folders. Recognising a filename does not mean every historical/client-specific variant can be rewritten by every tool.

Glossary

Allocation: the existing range of records a format exposes. A compatible free body block can be larger than one visible slot.

Art ID: the number selecting item artwork and its metadata, distinct from a character body or gump ID.

CHARDEF / ITEMDEF: Sphere character and item definition sections. A named definition can inherit from an existing numeric base using supported links.

DEFNAME: a stable script symbol used to refer to content by name.

Gump: UO interface artwork, or informally a complete dialog using that artwork. Studio separates the artwork browser from the dialog designer.

Hue / Partial hue: a palette transformation; partial hue limits colouring to applicable grey pixels. A preview toggle is not necessarily a saved item flag.

Layer: an equipment slot on a character, or an independently composited image in the layered preview. Context determines which meaning applies.

Mapping: a file/relationship directing a logical ID to its actual asset or bank. Changing a raw record does not necessarily change the mapping that selects it.

Multi: a native collection of item components used for a fixed structure such as a house. Functional door/sign objects can be supplied separately by server definitions.

Origin: the registration point used to align frames with their ground/body anchor. Stable origins reduce sliding and jumping between frames.

Paperdoll: the equipment view of a character in the UO interface. Its clothing images are separate from moving world-character art.

Queue / reviewed plan: a snapshot of prepared changes waiting for a game-file save, with dependency and conflict checks.

Recovery journal: the information and originals used to recover a linked save. Keep its complete folder intact.

Session checkpoint: stored editing state; it does not install the reviewed queue into game files.

Static / animated static: item artwork placed in the world; animated statics use frame metadata rather than a creature-body action layout.

TileData: client metadata describing properties such as art name, height, flags and equipment links. Sphere can supply additional or overriding behaviour.

What this release does not promise

The World Generator creates separate wilderness maps and does not edit an existing map in place or create server regions, NPC spawns, towns or bridges.

Version 0.30.1 does not contain a complete Sphere interpreter, full UO game-client emulator, networked CentrED service or general world-map editor. Its visual scenes do not establish server gameplay correctness or execute

attached sound events. File-access checks reduce save conflicts but cannot detect every program that has loaded data or prevent all later file opens. The house tool targets fixed compiled houses, not every custom-housing protocol or automatic migration of occupied houses.

Static reference and free-space checks do not resolve every dynamic expression, saved world object or client override. Modern UOP animation does not automatically convert to complete MUL bodies. Ordinary imports do not grow every allocation; the equipment expansion option is a specific reviewed capability. Content packs do not collect every dependency transitively or remap compiled houses.

32. World Generator: create a separate wilderness

Open **All tools > World Generator...**, or choose its task card on Home. The generator builds a new landscape from the selected client's existing terrain and scenery artwork. You can create a whole supported facet, preview it inside Studio and save its files separately. The linked client and server files are not replaced.

First generated world

1. Choose Small test world to explore the controls quickly, or select the intended full map size.
2. Choose the Map number that the exported files will use. Check Width and Height before generating.
3. Enter a World seed. The same seed and settings reproduce the same planned landscape. New random seed gives another starting point.
4. Choose Land shape, Sea %, Mountains, Climate and Forest %.
5. Set river, lake, volcano and path-link targets, then tree density, tree size and foliage density. Choose the reserved Black area, dungeon/cave counts and Interior size.
6. Choose Generate world. The progress window can cancel generation.
7. Click a point in the overview to inspect its actual UO terrain and scenery. Use X, Y and Go to inspect a known position. Choose Coast, Forest, Desert, Snow, Mountain, Volcano or Path beside Find area to jump to a matching region. Repeated clicks visit another nearby area. A message explains when that terrain is absent.
8. Adjust the controls and generate again, or choose Export new map folder when the preview is ready.

The UO terrain preview shows a local 21 by 21 tile area with the linked client's artwork, terrain heights and planned scenery. It is a visual inspection tool; it does not run a game server or prove that every tile will behave identically with every shard's movement rules. The world overview shows the complete generated regions. A white marker locates the area being inspected.

Explore the world in a separate window

After generating, choose Explore world in a new window above the preview tabs. This opens a separate isometric view of the generated landscape. Resize it or choose Maximise / restore to use more screen space. The window starts at the location selected in the generator. Closing it leaves the generator and its world available.

Control	What it does
Drag the map with any mouse button	Pan across the landscape

Control	What it does
Mouse wheel or Zoom	Choose 25%, 50%, 75%, 100%, 150% or 200%
Arrow keys or WASD	Travel in the corresponding screen direction while the map has focus
Shift with movement keys	Move three times as far per step
X, Y and Go	Travel to a tile coordinate inside this world
Click World map	Jump to that location in the whole-world minimap
Jump to a feature	Visit a volcano, lake, clearing, entrance, interior or reserved black area
Follow entrance / return	After choosing an entrance/interior, inspect the other end of its connection
Scenery	Show or hide trees, foliage, shoreline pieces and interior walls
Tile grid	Show terrain tile edges
Home or Starting location	Return to the initial location at 100% zoom

Click the main map before using movement keys. While typing coordinates, arrow keys remain available for editing the field. The minimap's white outline shows the approximate visible area, and the marker shows the view centre. The status line gives the centre coordinates, terrain height and zoom. At a map edge, empty space beyond the generated world is expected.

Only nearby tiles are drawn. Movement requests replace older unfinished views, so large maps remain navigable. Trees, shoreline pieces and ground selection use the same planned placement as export. The view remains an approximation of client rendering and is useful for inspecting slopes, shores, scenery spacing and routes.

The explorer displays the last generated world. Changing generator controls alone does not change it; Generate world refreshes an open explorer when generation succeeds. Opening different saved settings closes the old explorer. Pan, zoom, grid and scenery controls only affect the display. They do not edit terrain, export files, move a player character or connect to a server.

Size, map number and seed

Control	Meaning
Small test world	512 by 512 tiles for quick experiments
Britannia	7168 by 4096 tiles
Britannia (legacy)	6144 by 4096 tiles
Ilshenar	2304 by 1600 tiles
Malas	2560 by 2048 tiles
Tokuno	1448 by 1448 tiles
Ter Mur	1280 by 4096 tiles
Custom width and height	Multiples of 8, from 64 to 8192 per side, with at most 32 million tiles
Map number	0 through 5; controls the names of the exported map, statics and index files
World seed	A name or number of 1 to 80 characters used to reproduce the result

Map dimensions and map number are separate choices. Selecting a size does not configure your server or change its existing maps. A small test export needs a separate test setup with matching dimensions; do not substitute it for a larger facet without configuring that setup.

Generator version 2 improves river routing, filters tiny biome flecks and prevents crowded trunks. Small and medium trees stay at least four tiles apart; large trees need six tiles. Old version-1 settings still open, but regenerate using the current rules, so their new output can differ. Existing exported maps remain unchanged.

When controls change, the status says that the preview still shows the last generated world. Generate world updates it and enables export again. Find area and World Explorer inspect the currently generated snapshot.

The generator version, settings, painted regions and client artwork all matter for exact reproduction. A later generator version or a client with different native art can produce a different result. Keep the saved settings and exported native files together when archiving a finished world.

Height, slope and terrain review

After generation, use View above the World overview to switch between Biomes, Height, Slope and Shoreline / blends. Click any location to inspect it using the native UO artwork.

View	Meaning
Biomes	The normal landscape and region colours
Height	Dark lowland through light peaks; blue water stays at zero; reserved space stays black
Slope	Flat terrain is dark, smaller height steps are green, ordinary-limit steps are amber, and steeper rock faces are purple

View	Meaning
Shoreline / blends	Cyan shores, purple corners with three or more materials, red tiles without matching shoreline artwork in this client

Terrain review checks every native tile and shows exact counts together with placed/requested rivers, lakes, volcanoes, paths, dungeons and caves. It lists up to 24 spread-out examples per inspection category. Double-click a row, or choose Inspect selected location, to open that coordinate in UO terrain preview.

Terrain junctions and steps at the allowed limit are places to inspect, not automatic failures. The review does not simulate player collision, NPC pathfinding or server region rules. Counts describe the last generated world; changing controls still requires Generate world before export. The three inspection images and sample coordinates are included with exported maps.

Terrain restrictions and logical regions

Mountains setting	Maximum height	Ordinary ground step	Classic UO rock-face step
Gentle	28	2	8
Rolling	52	3	12
Rugged	80	4	16

Mountain edges offers Classic UO or Soft slopes. New worlds use Classic UO: rock interiors rise into craggy faces, with uneven capped ridges and graded margins. Soft slopes retains the ordinary-ground limit everywhere; older projects without this setting open with Soft slopes. Every horizontal and vertical terrain edge is checked before export. Larger steps are allowed only on edges touching rock. Water remains level at zero. A result exceeding these limits cannot be exported.

Raised terrain uses the client's square terrain textures when texidx.mul and texmaps.mul are available, including TexTerr.def mappings. Flat ground retains its diamond artwork. Missing optional textures fall back to that artwork. This improves mountain faces and their grass transitions; the preview remains an approximation of native client lighting.

Climate connects temperature with latitude and elevation. Dry regions favour warmer land, and snowy regions favour cold or high areas. Coastlines receive sand shelves. Snow receives earth margins, and craters receive rock margins. Ground brushes select matching native transition tiles between supported terrain pairs.

Sea % is a target from 25 to 80, and Forest % is a target from 0 to 85. Actual coverage changes when islands, coasts, mountain cores, intermediate ground and protected features are applied. These controls are not promises that exactly that percentage of the final tiles will have one appearance.

Rivers follow a drainage route toward the sea. The generator cuts level water channels and grades their banks instead of laying water uphill across the original slope. Lakes connect into the drainage network. Water surfaces use height zero. Volcanoes need enough surrounding land for their rocky slopes; a small, cold or fragmented island world may have fewer volcanoes than requested.

Rivers, lakes, volcanoes and paths

Rivers accept a target of 0 to 24, lakes 0 to 12, volcanoes 0 to 8, and path links 0 to 16. The result summary gives the actual number successfully placed. A requested feature is skipped when there is no suitable location or route. Try another seed, a larger landmass or less sea when a feature does not fit.

Paths connect suitable clearings across reachable land. They avoid water, lava and mountain cores and favour gentler terrain. This release does not construct bridges, towns or buildings. A path target of zero also disables the clearing network. River and path placement is part of generation, not a script installed in Sphere.

Generator version 3 considers every earlier clearing when seeking a reachable connection. A nearby clearing across water no longer blocks a possible connection elsewhere. Paths avoid diagonal gaps between blocked shore tiles. Version-1 and version-2 settings still open, but regenerate using current rules; preserve an older map's native export if you need its exact layout.

Reserved black area, dungeons and caves

Under Dungeons and caves, Black area reserves the right edge of the map for interiors and unused space. Small uses approximately one eighth of the width, Medium one quarter, and Large three eighths. At least 64 tiles remain for the outdoor region. New worlds start with Medium, two dungeons and two caves; older saved projects retain their original outdoor-only settings. Sea % applies to the outdoor region. A reserved world needs a width of at least 72 tiles.

Dungeons and Natural caves accept targets from 0 to 12 each. Interior size selects Small (48 tiles), Medium (80) or Large (128) planning areas. Stone dungeons have connected chambers and corridors. Natural caves have rounded chambers and wider passages. Native walls surround the floor, and black space separates the layouts. Interior floors are level at height zero.

Each placed interior receives a dry entrance on the outdoor map, preferably at a mountain edge. Natural caves use native black cave-mouth artwork with rough rock edges and a clear dirt approach. Their entrance trigger is immediately in front of the blocking artwork, linking to the separate interior rather than a walkable tunnel through the outdoor mountain. Stone dungeons retain an open entrance chamber and central trigger. Its inside return trigger is two tiles north of the arrival point. Landing points are separated from triggers so arriving does not immediately send the player back. Entrance platforms grade into the surrounding terrain and keep water level.

Requested counts are targets. When the black area is too narrow, the selected layouts do not fit, or there is insufficient dry land, the status message reports the actual number. Increase the map or reserved area, choose smaller interiors, lower the counts, reduce sea coverage or try another seed. Unused space stays black. Biome painting cannot replace it. Set both counts to zero to reserve an empty area for later map editing; also set Black area to None for a completely outdoor map.

In the separate explorer, choose a named entrance or interior in Jump to a feature. Follow entrance / return switches to its other end. Interior jumps choose 50% zoom. Use the wheel to see the whole layout, and switch Scenery off to inspect floor routes without walls. This is an in-app camera preview of the connection, not a running server.

Trees, foliage and saved biome regions

Tree density and Foliage each offer None, Sparse, Medium and Dense. Tree size offers Small, Medium, Large and Mixed. Size and density are independent: Medium trees with Sparse density gives occasional medium trees. Complete tree groups keep the correct canopy with their trunk; separate fragments are not scattered as if they were whole trees.

Vegetation varies with the ground region. Trees and plants are kept away from paths, shorelines, water, lava and mountain cores. Increasing foliage changes scenery density without raising terrain or changing the cliff limits.

Choose a biome under Paint biome regions, set Brush radius in tiles, then click the overview. The radius can be 4 to 2048 tiles. Undo last region removes the most recent stroke. Inspect switches back to selecting preview locations. Painted biome regions stay when the seed changes, allowing you to keep a desired desert or forest in the same broad area while changing the generated landscape. Resizing an already generated world scales its saved brush positions with the map.

A painted region changes the surface biome, not the elevation. Water, shoreline shelves, paths and crater cores remain protected. Regions do not lock every underlying terrain tile, river or mountain in place. The generator applies its normal intermediate ground around painted regions, so an abrupt snow-to-desert brush stroke still receives a buffer.

Save world settings writes a .uoworld.json containing the controls and painted regions. Open world settings reloads it and generates the preview. Save settings is separate from exporting native files. If you change a control after generating, Export asks you to generate the updated preview first, ensuring the exported result matches what you have reviewed.

Export contents and checks

Choose Export new map folder and select an existing folder outside your linked client and server. Studio checks required source files for open handles, destination write access, free disk space and terrain limits. It writes into a temporary folder, verifies native block sizes and scenery indexes, then publishes one uniquely named result folder. Cancellation removes that operation's temporary output. Existing files in the selected output directory are not overwritten.

Exported file	Purpose
mapN.mul	Native terrain tile IDs and signed heights
staticsN.mul	Trees, foliage, shore pieces and interior walls
staidxN.mul	Native index locating each scenery block
overview.png	Complete world overview
terrain-height.png, terrain-slope.png	Height and slope inspection views
terrain-shoreline-review.png	Shoreline, mixed-material and missing-artwork markers
heightmap.bmp	Grayscale height plus 128, compatible with CentrED height import
world.uoworld.json	Reproducible seed, controls and saved biome strokes
world-report.json	Dimensions, height checks, feature counts, scenery totals and terrain-review sample coordinates
dungeon-links.json	Entrance, exit and landing coordinates when interiors exist
server-scripts/maps/generated	Companion Sphere entrance-link script when interiors exist
READ-ME.txt	File-set and separate testing instructions
SHA256SUMS.txt	Checksums for the completed output files

Keep the terrain, statics and index together. A client requiring mapLegacyMUL.uop can use All tools > MUL to UOP on the generated mapN.mul. Statics and their index remain separate. The converter uses directly readable terrain chunks and the native internal entry names for alternate map files.

Activate entrance links in a separate test server

When interiors exist, keep dungeon-links.json and the companion server-scripts/maps/generated script with the exported map. The script uses Sphere's [Teleporters] section for entrance and return coordinates. Review its map number and positions, copy it to scripts/maps/generated in a separate test server, and add its path to that server's spheretables.scp include list. Load the matching map, statics and index with the same facet dimensions. Map files alone do not activate these links.

Walk through every entrance and return point in that separate setup. Check wall corners, collision, the clear approach and landing positions using your own client/server rules. Studio does not install the script or change live files. No creatures, loot, lighting regions or server behaviour are supplied by the dungeon layout.

Further editing and deployment

For further editing in CentrED, create a separate workspace containing the generated native file set and your own matching client artwork. Use the exact tile dimensions from the report. CentrED's block dimensions are the tile dimensions divided by eight. The exported height image uses the height-plus-128 convention expected by its grayscale height importer.

Inspect coastline corners, routes, mountain approaches and biome junctions before deploying a world. Stock single-tile brushes cannot represent every three-material junction; the report counts these locations for final editor polish. The preview uses native artwork but is not a complete client renderer or movement simulator.

The world-report.json export also records generator_version and unmatched_coast_tiles. A nonzero unmatched count means the chosen client lacks matching shore artwork for some corners; the completion status calls this out. Inspect the shores before using the map. The report also counts three-material junctions that may need final map-editor polish.

World Generator creates outdoor terrain, optional interiors and companion entrance-link scripts. It does not create NPC spawns, towns, server regions, housing deeds, player start locations or a new server configuration. Existing map patches, regions and saved object positions belong to the old world and need separate review. Test matching client and server copies before replacing any live map; exporting from Studio performs no deployment.

33. Map Regions: boundaries and server rules

Open existing regions

Choose Map Regions in the sidebar, or All tools > Worlds > Map Regions. This editor targets Sphere AREADEF and ROOMDEF scripts. RunUO, ServUO and ModernUO region formats are not edited by this release.

Search the left list by name, identifier or source filename. A dot marks a changed draft. Click a name or map boundary to select it. Overlapping map boundaries select the smallest matching region; use the list for an enclosing region. Find selected zooms to its rectangles.

The editor reads regions from the project's script collection, including files which may not be active in spheretables.scp. It does not run the server or resolve every custom event. Region ordering and overlapping rules still follow the shard's Sphere configuration.

Load and navigate a terrain map

The editor attempts to read map0.mul or map0LegacyMUL.uop from the linked client when it opens. Load terrain lets you choose a different MUL or legacy map UOP, including a separately generated world. Select the corresponding map ID. Standard facet sizes are detected and loaded automatically, including 6144 x 4096 and 7168 x 4096 Britannia maps. The loaded dimensions appear above the preview.

Custom maps require their actual width and height in tiles, divisible by eight. UOP files do not store width and height separately, so Studio recognises standard sizes from the terrain length. Cancelling the dimensions window keeps the current preview and scale. Version 0.30.1 fixes the incorrect dimension error for standard legacy UOP archives, including their optional trailing padding block; you do not need to convert the UOP to MUL first.

A sibling radarcol.mul supplies terrain colours. Without it, the map uses a labelled grayscale elevation preview. This is a terrain overview: it does not render buildings, statics or live characters. Files are opened for reading, and loading a map does not write client files. Changing the map ID clears the background so boundaries are not drawn over the previous facet by mistake.

Wheel to zoom, middle-drag to pan, or choose Fit map. Without terrain, edit boundaries on the coordinate grid. The status line shows the pointer coordinates.

Adjust, add or remove boundaries

1. Select an existing region, or choose Add region for a new AREADEF with a unique identifier.
2. Enter the display name in Region details.
3. Open the Boundaries tab, choose Draw boundary and drag across the map. The first draw replaces a new region's placeholder; further draws add rectangles to the same region.
4. Select a rectangle in Boundary rectangles. Edit X1, Y1, X2 and Y2, then choose Update boundary. Coordinates are decimal; X2 and Y2 are the first tiles outside the boundary.
5. Choose Move boundary and drag to move the selected rectangle without changing its size. Movement is constrained to the displayed map.
6. Remove boundary removes one rectangle. Every retained region needs at least one rectangle; Remove region removes the whole region from the draft.

A region can contain up to 256 rectangles, all on one map. Irregular areas are represented by several rectangles. Names and boundaries must pass validation. Regions with expressions rather than literal boundaries are marked for editing in the script editor instead.

Apply to draft records the current details locally; selecting another region and opening Review changes also apply valid details. Undo restores up to 50 earlier draft states. Closing an edited draft asks whether to discard it. Unreviewed region drafts are not included in session checkpoints, so review and queue work you want to keep between sessions.

Choose the rules

Open the Rules tab in Region details.

Control	Sphere behaviour
Guarded	Enables the guarded region flag; custom guard events and configuration still apply
Safe from harm	Marks the region safe; this takes precedence over guarded behaviour
Allow spells	Clears the global antimagic restriction when enabled
Allow harmful spells	Controls the harmful-spell antimagic restriction
Allow recall / mark into region	Controls the recall-in / mark restriction
Allow recall out	Controls the recall-out restriction
Allow gate travel / Allow teleport	Controls the separate gate and teleport restrictions
Allow PvP / Allow building	Controls player combat and building restrictions
Underground / no weather	Marks the region underground
Announce entry / Instant logout	Sets the entry-announcement or instant-logout flags
Prevent item decay / Arena rules	Sets the corresponding Sphere region flags

Checked means the named rule is enabled. Disabling Allow spells overrides the more specific travel permissions; recall restrictions can also affect Mark and Gate in Sphere. These are region flags, not an arbitrary individual spell blacklist. Custom triggers and server settings can impose further restrictions.

Recognised numeric and standard flag expressions populate these controls. Unknown flag bits are retained when known rules change. A custom expression that Studio cannot resolve makes the rule controls read-only; its original text is retained while supported name/boundary changes remain available. Edit such flags in Sphere scripts when you know the shard-specific expression.

Review, queue and save

Choose Review changes. Save or discard any open script edits first. The review shows the exact script differences, new files and resource-list changes. Original unrelated sections, custom tags, events and the order of retained regions are preserved; existing script encoding and newline format are retained.

New regions go in a uniquely named file under scripts/regions and are registered in spheretables.scp. A new region contains a name, centre point, boundaries and chosen flags. It does not automatically copy custom resource events, spawn rules, guards or scripts from another region. Configure these through the script editor where the shard requires them.

The backup option starts enabled. Queue changes adds the reviewed operation to Pending changes; Save All writes the queued work. Saving immediately uses the same access checks and recovery journal. A changed script collection stops the reviewed operation and requires a fresh region review.

Stop Sphere before saving, then restart it. Keep a covering world region; without one, parts of the map can become unusable. Region removal leaves terrain, objects and saves intact. Verify spell, travel, guard and overlap behaviour on your shard; the overview does not simulate server rules.