

1.5.0 Change Log

Primary Changes

- 1) The first attempt at **MicroVerse** integration has been made. Visit the [MicroVerse](#) page in our Docs for more information!
- 2) Removed the **Initialize On Awake** option from the **Component Manager**, as well as the Context Menu options allowing you to initialize it from the editor. This functionality has been moved to the **SAM Initializer**, and it is through this component that all initialization should occur now (unless you wish to use a custom component to achieve similar functionality).

To duplicate the behavior of **Initialize On Awake** using the SAM Initializer, enable the **Initialize On Startup** option and set the **Initialize Type** option to **Two_Frame**.

- 3) Added new fields to the **SAM Initializer** to allow you to set game objects to be destroyed before/after initialization. Also modified the order fields are displayed in the SAM Initializer inspector to better show the order the actions are executed at runtime.

Finally, you can set the SAM Initializer **to Initialize On Startup**, and also choose whether it initializes the Component Manager in a gradual way or using two frame initialization (the former will be triggered from the SAM Initializer's Start method, the latter from its Awake method).

- 4) Added a new option to the World Designer Tool's **What To Assign** dropdown called **Game Object With Component** (reminder, this dropdown is now found in the **Advanced Operation Settings** controls, in the **Assignment Settings** tab). Choosing this option allows you to assign game objects during **Assignment Operations** only if they contain a specific component.
- 5) Added a new option to the World Designer Tool under the **General Settings** tab of the **Advanced Operation Settings** controls, called **Try Use Mesh Bounds**. When enabled, this will force the World Designer tool to use the center of each assigned/transferred object's MeshRenderer Bounds to calculate the Cell that the object should be assigned/transferred to. If no MeshRenderer exists, it defaults to using the Transform position of the object, which is the same as the old behavior and the behavior when this option is disabled.

This is very useful for situations where the transform position of the object does not accurately reflect its position in the scene or in situations where the meshes are not symmetrical.

- 6) Added support for using scenes with activated root game objects (previously SAM required all scenes to use deactivated root game objects). This change was made primarily to support Unity's Build Time Static Batching functionality, which requires the root objects to be activated. More details can be found in the [Static Batching Section](#) of the FAQ Chapter in our online documentation.

- 7) The Scene Formatter Editor Window functionality has been moved to a Scriptable Object, which you can create using **Create -> Deep Space Labs -> SAM -> Scene Formatter**.

This Scene Formatter asset now has an option (called **Deactivate Roots**) to specify whether the root objects in the formatted scenes should be active or deactivated.

This setting defaults to enabled (meaning the roots will be placed in a deactivated state), as this is the normal state of scene root objects that works better with SAM.

This option was solely added to support Build Time Static Batching, although it might be useful for other things.

- 8) Added a new **Advanced Data Set Methods** dropdown to the Streamable Grid inspector (found at the bottom of the **Global Main Settings** tab). This dropdown currently includes a single new data set method, which allows you to enable Cells by searching a specific folder for the presence of Asset Chunks associated with LOD Group 1 of the Streamable Grid (any asset form will do whether it be prefabs, scenes, or some other format). The operation only enables cells (it does not disable cells), and is very useful for game developers who are importing their Asset Chunks directly or using Asset Chunks generated using 3rd party tools.
- 9) The Standard Hierarchy World Shifter will now execute PlayerMover related code in the manner dictated by the **When To Shift** setting.

10) When using a **Per Material Transitioner**, you can now enable the **Add Material Destroyer** option (only shown/used if **Use Shared Materials** is disabled), and the Transitioner will add to each transitioned game object a component which will destroy the Material instances created by the Transitioner in the event that the Game Object is destroyed.

This can be used as an alternative to destroying the Materials manually (which is a technique that is unlikely to be used) or calling **Resources.UnloadUnusedAssets** (more likely). We recommend testing each method to weigh the cost of the increased memory usage of the Material Destroyer vs the (assumedly) worse performance of **Resources.UnloadUnusedAssets**.

Fixes

- 1) Fixed various errors/exceptions/issues with resizing Streamable Grids to large values, and also added better error handling for instances when a resize operation cannot be executed (due to it resulting in internal collection memory size growing too large). Also improved the speed of some of these operations (especially when working with large Streamable Grids).
- 2) Fixed some issues with the Loading Blueprint Editor when working with large Streamable Grid sizes.
- 3) Fixed an exception that could occur when loading a SAM main scene from another scene, when the SAM scene uses two frame initialization and a Scene Chunk Streamer or Addressable Scene Chunk Streamer.
- 4) Fixed the following error message in Unity 2020.3 version: **The name 'PrefabStageUtility' does not exist in the current context.**
- 5) Fixed a **NullReferenceException** that could occur at runtime if using a Chunk Type of None.
- 6) Fixed an **ArgumentException** that could occur when pressing the World Designer Tab on the World component inspector.

Other

- 1) You can now use alternative transition settings for LOD Transitions when using **Cell Visual Transition Controllers**. This allows you to have different transition settings than regular non LOD Transitions, which is useful in some scenarios (such as when using cross-fading).
- 2) When you set the rows or columns of the Streamable Grid to a value greater than **999,999**, the scroll bar on the Visual Grid is now replaced with new buttons that allow you to increment/decrement the rows and columns by 1 or 4 rows/columns at a time. This is needed because the scroll bar does not work correctly for very large numbers.
- 3) The size of the Streamable Grid's Visual Grid buttons increases as larger and larger numbers are used. This allows the buttons that show the row, column, and layer numbers to fit the entirety of those larger numbers.
- 4) New input fields (**First Row To Show Below** and **First Column To Show Below**) have also been added above the Visual Grid so you can explicitly set which row/column is shown first among the 4 rows/columns that are shown (useful when working with larger Streamable Grids).
- 5) **Int Sliders** throughout SAM will now be replaced by Int Fields when the max slider number is greater than **999,999**. This change was required because the Int Slider GUI Control does not work correctly for extremely large numbers; however note the cutoff of 999,999 is largely arbitrary. This mostly affects the Streamable Grid inspector and World Designer Tool.
- 6) Removed the **Layer number** from Cells shown in the Loading Blueprint Editor and World Designer Tool in order to make room for more digits for the Row and Column Numbers (needed for large Streamable Grids mostly). The Layer number can still be found in the Cell Viewer where it has always appeared.
- 7) Made various improvements to the **World Designer Tool** to better accommodate large Streamable Grids. One such example is the **Viewable Area**. You will now need to press an **Update View Area** button to apply manually inputted start and end row/column/layer values. Previously, the manually inputted values would be applied immediately, which would work fine for smaller Streamable Grids, however for

larger grids where you may be working with values $\geq 10,000$, having the changes apply automatically could result in extremely large temporary viewable areas, which would crash the Tool.

- 8) Deactivated Worlds in the scene that are assigned as Component Manager Prototypes should now have their Asset Chunks removed when entering Play Mode, if those World Prototypes were used to load Asset Chunks via the World Designer Tool and those Asset Chunks were not removed prior to entering Play Mode.

Asset Chunks loaded using deactivated Worlds that are not assigned as Prototypes will not be removed, as it's assumed those chunks were left loaded on purpose for testing purposes (in those scenarios, if they are not meant to be loaded when entering Play Mode, you must manually unload them using the World Designer Tool).

- 9) Added Job based methods for finding the endless or streamable grid cells a batch of positions are in.

- 10) The World Grouping Name for a Grouping will now automatically be set to the Streamable Grid name when assigning a Streamable Grid to the Grouping, so long as the Name is current blank or set to **Unnamed**.

- 11) Added a new package version for Unity 2023.1 and greater versions.

Online Docs/API

- 1) Clarified/Revised public API on existing Position Translation methods on the World class.
- 2) Added some new batch translation methods that can be used from methods driven by the World's update cycle (the previous methods were not suitable for that purpose). These methods have a return type of `IEnumerator<YieldInstruction>` and `JobHandle`, whereas the existing methods have a return type of `void`.
- 3) Added batch [FindEndlessGridCellsPositionsAreIn](#) and [FindStreamableGridCellsPositionsAreIn](#) methods to the World class, which use Jobs to speedily find the cells positions are in.

- 4) Removed the dedicated Scene Formatter Chapter and moved the [Scene Formatter](#) section from that chapter to the Scriptable Assets Chapter.
- 5) Added new [Static Batching](#) and [SRP Batcher](#) Sections to the FAQ Chapter of the Docs.
- 6) Added a new Third Party Tools Chapter and a Section on [MicroVerse](#) to it.