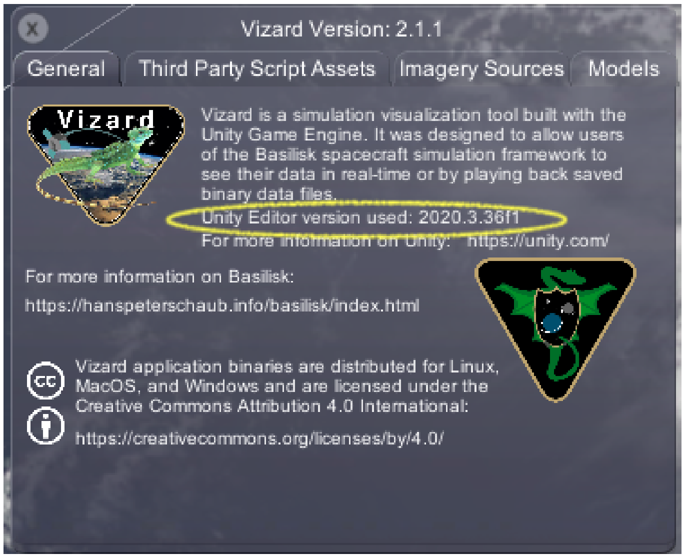
How to Build Custom Unity Addressables Bundles for Vizard

Version 1.0

1. Open the VizardContent Unity project. Unity states that the Unity editor version used to generate the VizardContent Addressables bundle must match the Unity editor version (i.e. Unity 2020.3) used to build the version of Vizard you are using.

   The Unity Editor version used can be found by looking at the Vizard app's info or by opening the "About Vizard" panel (File>About Vizard) while playing back a file.

   Please note that user-generated Addressables are not supported in Vizard versions earlier than Vizard 2.1.1.



Vizard Version: 2.1.1

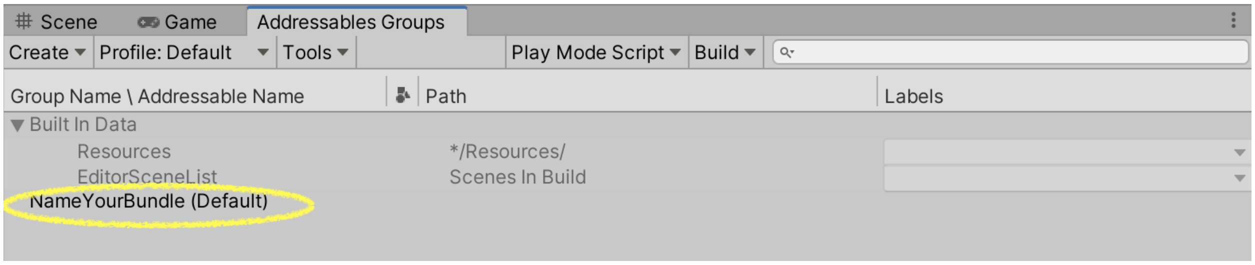
General | Third Party Script Assets | Imagery Sources | Models

Vizard is a simulation visualization tool built with the Unity Game Engine. It was designed to allow users of the Basilisk spacecraft simulation framework to see their data in real-time or by playing back saved binary data files.
Unity Editor version used: 2020.3.36f1
For more information on Unity: https://unity.com/

For more information on Basilisk:
https://hanspeterschaub.info/basilisk/index.html

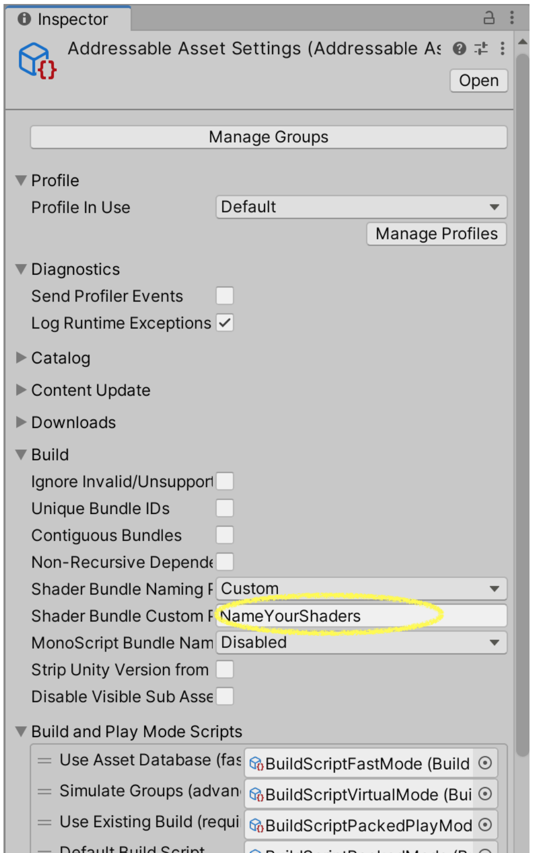
Vizard application binaries are distributed for Linux, MacOS, and Windows and are licensed under the Creative Commons Attribution 4.0 International:
https://creativecommons.org/licenses/by/4.0/

2. Open the Addressables Groups tab by clicking Window>AssetManagement>Addressables>Groups. We recommend you rename the Default group to a name specific to your bundle to help you keep track of your Addressables.



Group Name \ Addressable Name	Path	Labels
Built In Data		
Resources	*/Resources/	
EditorSceneList	Scenes In Build	
NameYourBundle (Default)		

3. Provide a custom name for the built-in shader bundle. This will prevent Unity having trouble with its built-in shaders if you have more than one bundle in your CustomModels folder. To change the built-in shader bundle name:
   a. In the search box below the main window, type "AddressableAssetSettings".
   b. Click on the search result and see the settings in the Inspector Tab
   c. Provide your own custom name in the "Shader Bundle Custom Prefix" field.
   d. DO NOT CHANGE ANY OTHER SETTINGS, ESPECIALLY THE BUILD AND LOAD PATHS



Profile
Profile In Use: Default

Diagnostics
Send Profiler Events ☐
Log Runtime Exceptions ☑

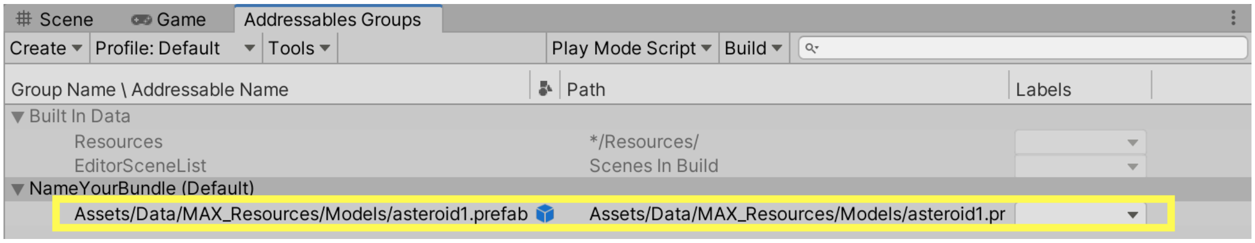
Catalog
Content Update
Downloads

Build
Ignore Invalid/Unsupport ☐
Unique Bundle IDs ☐
Contiguous Bundles ☐
Non-Recursive Depende ☐
Shader Bundle Naming P: Custom
Shader Bundle Custom P: NameYourShaders
MonoScript Bundle Nam: Disabled
Strip Unity Version from ☐
Disable Visible Sub Asse ☐

Build and Play Mode Scripts
Use Asset Database (fas: BuildScriptFastMode (Build
Simulate Groups (advan: BuildScriptVirtualMode (Bui
Use Existing Build (requi: BuildScriptPackedPlayMod
Default Build Script

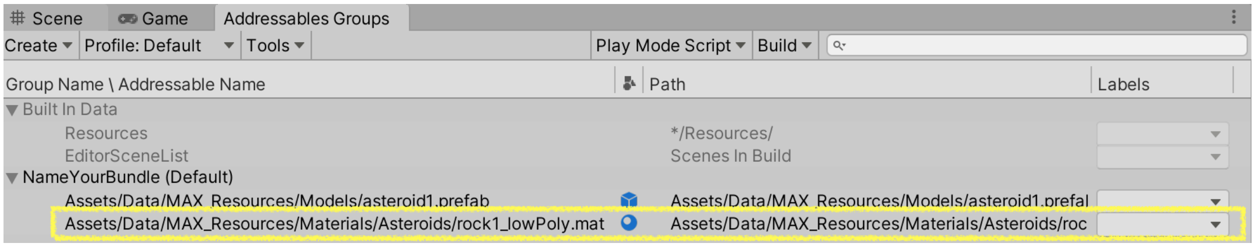
4. Click the “Addressable” checkbox in the top left of the Inspector of your prepared prefab object to add it to the bundle you are building. Note that the name of this prefab should exactly match the ModelKey you will be telling Vizard to load in your spacecraft or celestial body message. Details on how to prepare your model(s) for bundling are provided in the “Importing and Preparing Custom Models to Build” instructions.

   Note that it appears in the Addressables Groups window under your bundle’s name:



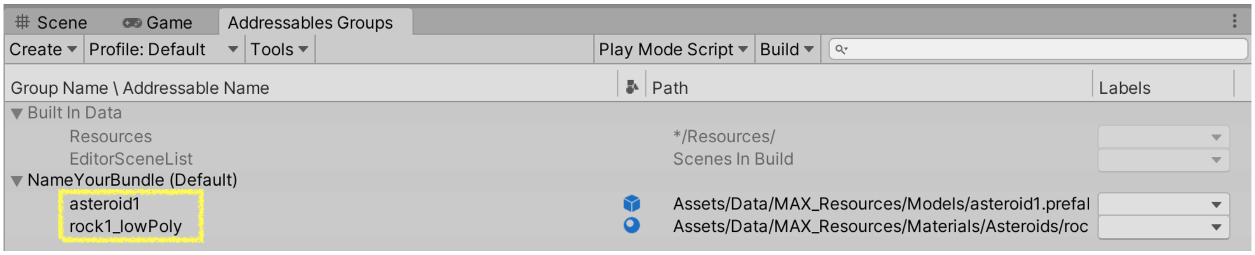
Group Name \ Addressable Name		Path	Labels
▼ Built In Data			
Resources		*/Resources/	▾
EditorSceneList		Scenes In Build	▾
▼ NameYourBundle (Default)			
Assets/Data/MAX_Resources/Models/asteroid1.prefab		Assets/Data/MAX_Resources/Models/asteroid1.pr	▾

5. Select any materials that should be included with this prefab. Click the “Addressable” checkbox in the top left of the Inspector tab for all of them. See that the materials also appear in the Addressables Groups window under your bundle’s name:



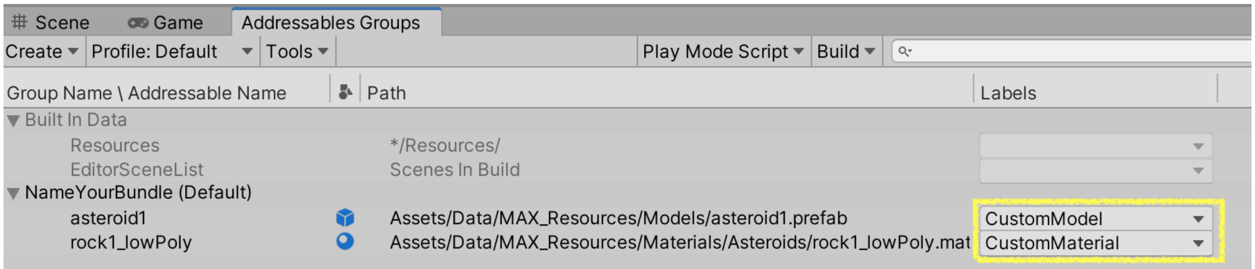
Group Name \ Addressable Name		Path	Labels
▼ Built In Data			
Resources		*/Resources/	▾
EditorSceneList		Scenes In Build	▾
▼ NameYourBundle (Default)			
Assets/Data/MAX_Resources/Models/asteroid1.prefab		Assets/Data/MAX_Resources/Models/asteroid1.prefal	▾
Assets/Data/MAX_Resources/Materials/Asteroids/rock1_lowPoly.mat		Assets/Data/MAX_Resources/Materials/Asteroids/roc	▾

6. Right click on Addressable Name of each of your bundled assets and select “Simplify Addressable Names”. The resulting simplified Addressable Name for your prefab should match the ModelKey you intend to use. This is important. It is how Vizard is going to find your model in the asset bundles.



Group Name \ Addressable Name		Path	Labels
▼ Built In Data			
Resources		*/Resources/	▾
EditorSceneList		Scenes In Build	▾
▼ NameYourBundle (Default)			
asteroid1		Assets/Data/MAX_Resources/Models/asteroid1.prefal	▾
rock1_lowPoly		Assets/Data/MAX_Resources/Materials/Asteroids/roc	▾

7. Please select the appropriate Label for each of your assets from the dropdown menu. Label the model prefabs “CustomModel”, label the materials “CustomMaterial”.



Group Name \ Addressable Name	Path	Labels
Built In Data		
Resources	*/Resources/	
EditorSceneList	Scenes In Build	
NameYourBundle (Default)		
asteroid1	Assets/Data/MAX_Resources/Models/asteroid1.prefab	CustomModel
rock1_lowPoly	Assets/Data/MAX_Resources/Materials/Asteroids/rock1_lowPoly.mat	CustomMaterial

8. Add as many Finalized Model Prefabs and their associated Finalized Materials as you would like to your bundle by repeating Steps 4 through 7 for each When they have all been marked Addressable, the models have had their names simplified to their ModelKey, and the assets have been labeled as CustomModels or CustomMaterials you are ready to export your asset bundle.

EXPORTING YOUR ASSET BUNDLE TO USE WITH VIZARD

1. Set the Target Platform
   The VizardContent project to the platform you will be running Vizard on (MacOS, Windows, or Linux) by going to File>BuildSettings and setting the Target Platform. You can close the Built Settings window after you have set the platform. Note:
   a. Vizard is built to the Intel 64-bit + Apple Silicon architecture
   b. Vizard is built with the Mono backend in Linux to the x86_64 architecture
   c. Vizard is built to the x86_64 architecture in Windows

2. Build the Bundle
   In the Addressables Groups tab in the main window click on the “Build” dropdown at the top right, next to the search bar. Select NewBuild>DefaultBuildScript. This will start building your Addressables bundle.

3. Locate the Bundle and Rename Catalog Files (if desired)
   The finished bundle will be located in the VizardContent>Assets>Exported directory of the VizardContent project. Note that there are four files for each exported bundle:
   a. catalog.json file
   b. catalog.hash file
   c. your_bundle_name_assets.bundle
   d. your_bundle_name_Resources.bundle
   [Optional] You can rename the catalog.json and catalog.hash files. Make sure you use the same string for both (i.e. your_bundle_name.json and your_bundle_name.hash). This can help you keep track of which catalog files go with which bundles if you are maintaining more than one asset bundle.

4. Move the Bundle to Vizard's Application Support Library
   Move the four files associated with your exported bundle from the VizardContent>Assets>Exported directory to your Vizard application's application support library. (Don't worry about the associated .meta files you see, they are just Unity Editor bookkeeping and don't need to move in the next step.)

   Vizard 2.1.1 and later will try to automatically add the Resources/CustomModels/ subdirectories to your application support library if they are not already present (Note that the automatic folder creation failed with MacOS Monterrey and the user had to create the Resources/CustomModels/ subfolders manually).The paths vary by operating system:
   a. MacOS:
      Users/your_user_name/Library/Application Support/Vizard/Vizard/Resources/CustomModels
   b. Windows:
      C:/Users/your_user_name/AppData/LocalLow/Vizard/Vizard/Resources/CustomModels
   c. Linux (Mint): home/your_user_name/.config/unity3d/Vizard/Vizard/Resources/CustomModels

5. Use the Models in Vizard
   You can now run Vizard and have it replace default models with the model you specify by setting the modelDictionaryKey of either the Spacecraft or CelestialBody message.

UPDATING OR ADDING TO YOUR BUNDLE

You can change your models and materials or add additional models and materials to the same bundle and export the bundle again. You will have to replace the older four files in the application support library with the newly generated set.

CREATING ADDITIONAL BUNDLES

If you want to have completely separate bundles with different assets and materials in each bundle, we have found this process to work:

1. Relocate or delete any old bundles in the VizardContent/Assets/Exported directory.
2. Uncheck the Addressables box for all the models and materials you DO NOT want to include in the new bundle.
3. Check the Addressables box for all the models and materials you DO want to include in the new bundle
4. Rename the Addressables Group AND the shader prefix to describe your new bundle
5. Select all your Addressable assets, right click and select Simplify Addressable Names to make sure your ModelKeys will work.
6. Assign the appropriate Label (CustomModel or CustomMaterial) to your Addressable Assets
7. Export your new bundle by clicking Build>NewBuild>DefaultBuildScript
8. You can place your new bundle in the Vizard/Vizard/Resources/CustomModels directory alongside other bundles you have made. Vizard can search through multiple bundle catalogs to find your model at runtime.

AVOID DUPLICATE MODEL KEYS

Vizard will search through available bundles and load the first model it finds with a model key that matches what was requested. It is case-sensitive. The requested model's key MUST match what is set in Basilisk for the modelDictionaryKey.

TROUBLESHOOTING

- Problem: Model loads, but is bright pink
  If your model loads in Vizard, but the material is showing bright pink, check that you had the correct platform selected under File>BuildSettings when you exported your Addressables bundle because the Shaders are platform-specific. You will have to fix the platform and export the bundle again.

- Problem: Can't rename the default group
  Try closing the VizardContent project and reopening it. Note, you can add groups if you want, but they will all be exported under one catalog and the shaders bundle will always take the name of the default group.