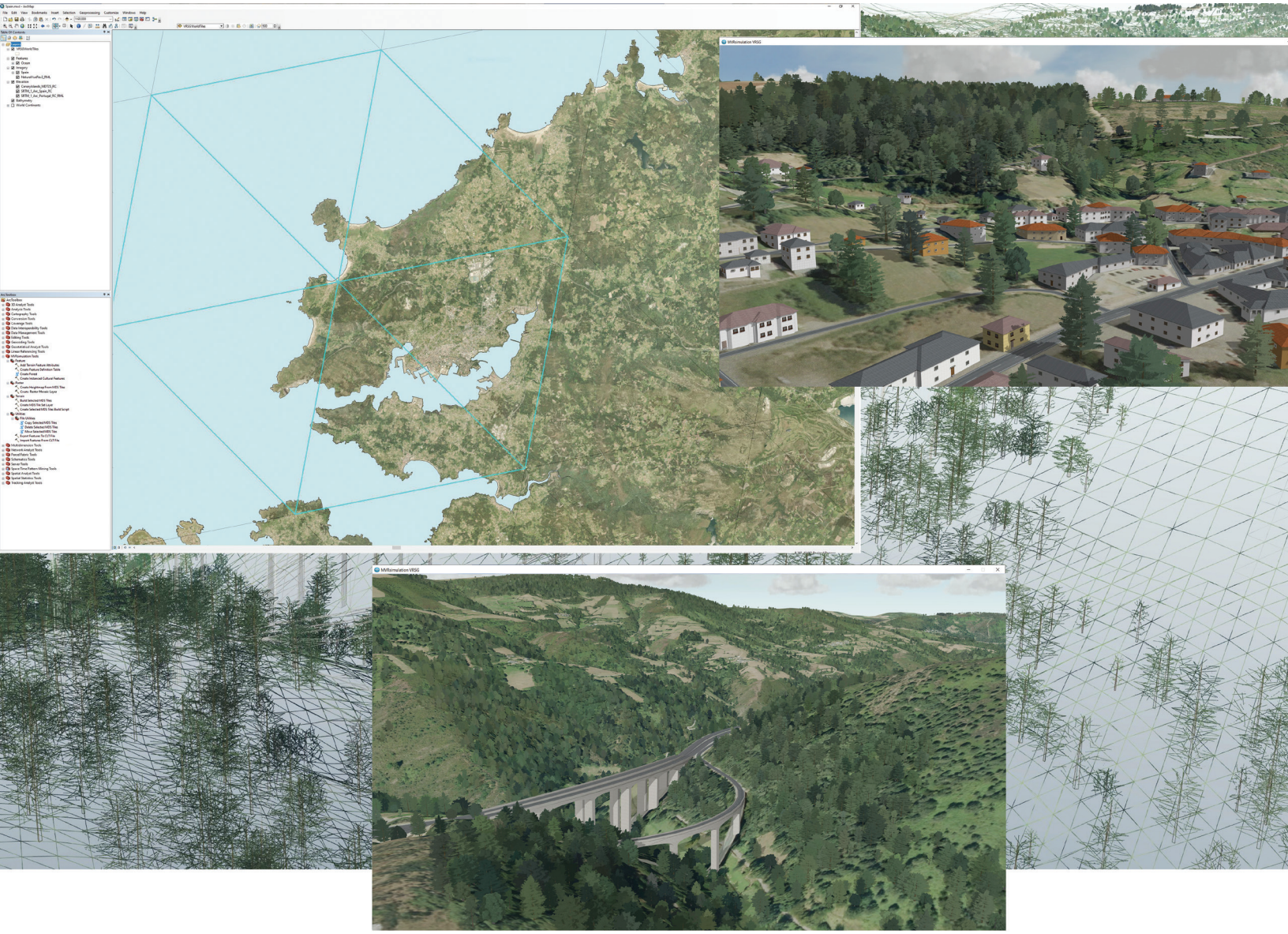
Terrain Tools for Esri ArcGIS 10

Version 2.0 User's Guide



Geospecific simulation with game quality graphics

MVRsimulation

Printed in the United States of America. Reprinted September 2023

MVRsimulation Inc.
57 Union Avenue
Sudbury, MA 01776 USA
Phone: 617-739-2667 Fax: 617-249-0151
Website: www. mvrsimulation.com Email: sales@mvrsimulation.com

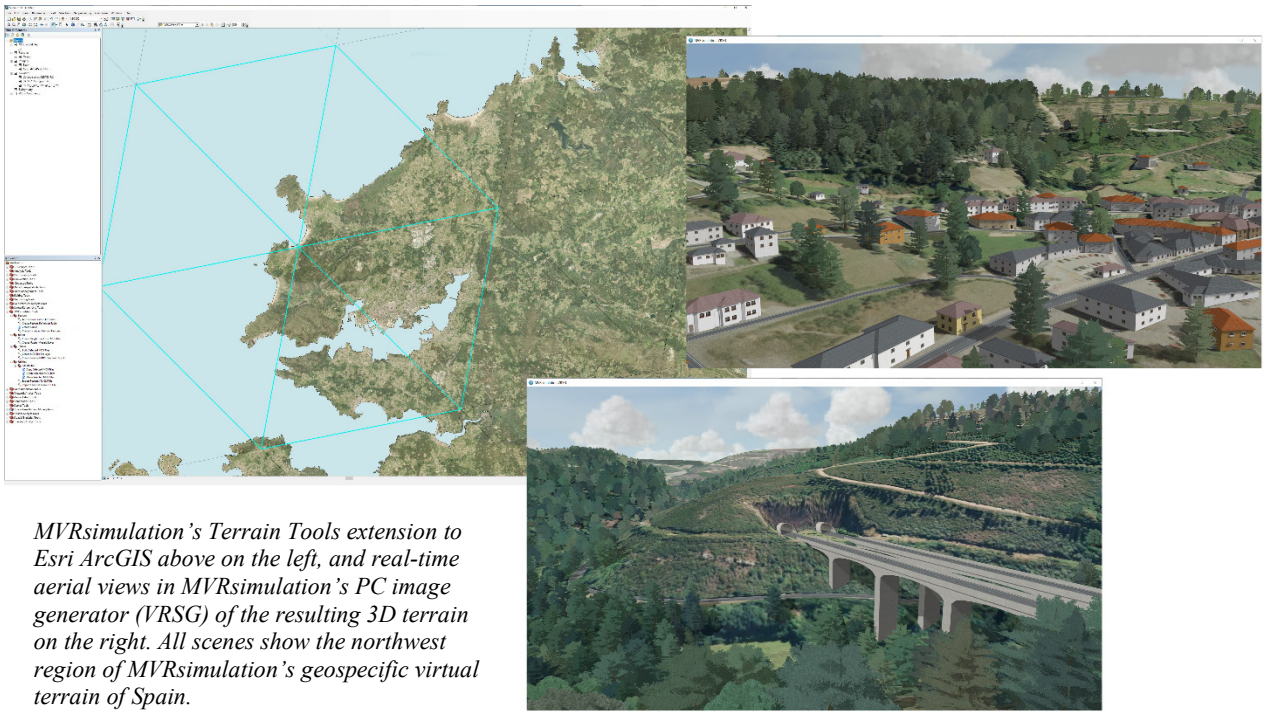
Contents

CHAPTER 1

Introduction

MVRsimulation Terrain Tools extension for Esri® ArcGIS® Desktop enables you to build real-time 3D virtual terrain using geospatial data as input. These terrain generation tools generate 3D terrain, for visualizing in MVRsimulation's Virtual Reality Scene Generator (VRSG®).

Leveraging the infrastructure of Geographic Information Systems (GIS) technology and building on the industry standard ArcGIS platform, MVRsimulation's Terrain Tools extension combines powerful 3D terrain building functionality with an accessible and intuitive interface. Terrain Tools can be easily understood by anyone with a general knowledge of ArcGIS.



MVRsimulation's Terrain Tools extension to Esri ArcGIS above on the left, and real-time aerial views in MVRsimulation's PC image generator (VRSG) of the resulting 3D terrain on the right. All scenes show the northwest region of MVRsimulation's geospecific virtual terrain of Spain.

The minimum amount of source data required to build virtual terrain for an area of interest is overhead (aerial or satellite) imagery and digital elevation data covering the area. You can also use vector data sources, such as geo-located polygons, in the terrain building process to refine input raster data or add terrain features to the output.

The Terrain Tools software is delivered with sample imagery, elevation, bathymetry, and vector data of downtown Baltimore, MD. You can use this data to become acquainted with using the tools to build 3D terrain. This sample source data can be found in the Terrain Tools installation directory (usually C:\MVRsimulation\ArcGIS Terrain Tools\Sample Project). This data is used in the tutorial exercises in the next chapter.

VRSG round-earth 3D terrain format

MVRsimulation's VRSG round-earth terrain format defines a tessellation of the Earth's surface into a set of triangular tiles, each of which is generated independently and stored in the file system as an MDS (.mds) file. Each MDS output file is known as a *terrain tile*.

The VRSG round-earth format is ideal for aerial applications, which require vast areas of 3D terrain. Terrain Tools is currently used to build terrain for many manned and unmanned aircraft simulators as well as for close air support training.

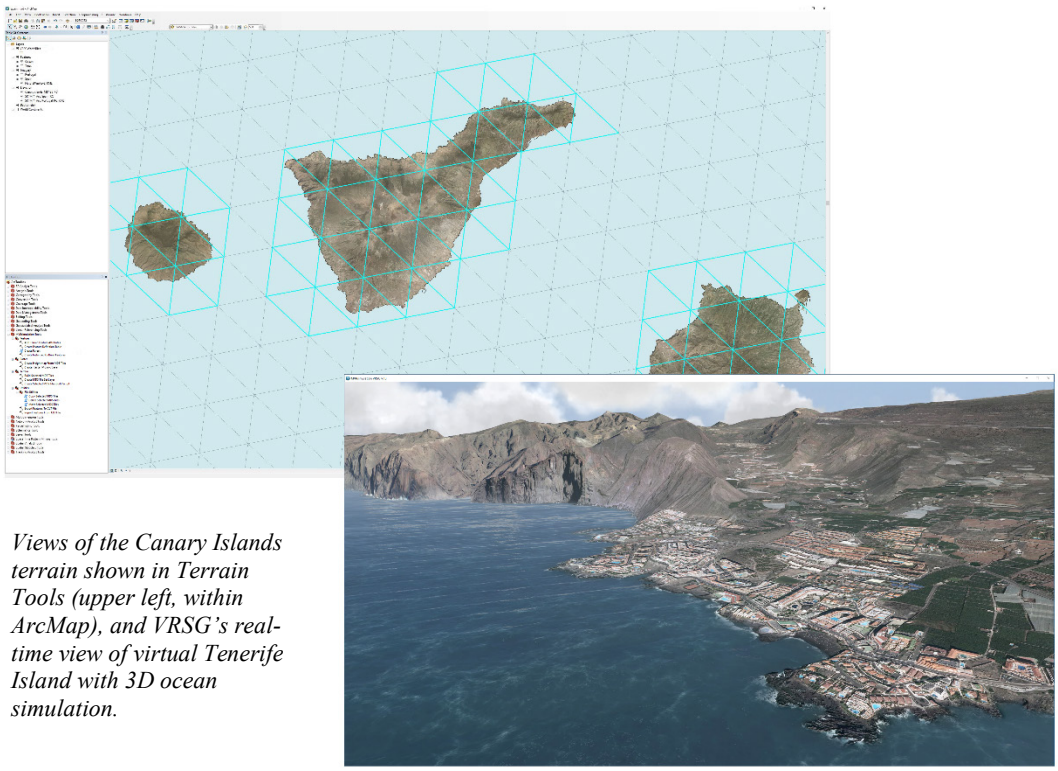
MVRsimulation's Terrain Tools and Virtual Reality Scene Generator (VRSG) real-time textured and wireframe 3D views of the virtual Albacete Air Base, Spain. Also shown is the air base's control tower model in MVRsimulation's Model Viewer.

The round-earth terrain format has many benefits; most importantly, the terrain models the earth to a high degree of accuracy over its entire surface in contrast to a local approximation that is only valid over a relatively small range. This level of accuracy is vital for targeting applications and determining intervisibility. Another critical advantage of the terrain format is its segmented database representation. Terrain built in the VRSG terrain format is comprised of relatively small, self-contained terrain tiles that fit together seamlessly but can be built

separately. This property enables a high degree of parallel processing of terrain since many different machines can build tiles for a given area at the same time.

Within the ArcGIS interface, coastlines can be digitized and cut out from geospecific imagery with Terrain Tools. At runtime, VRSG generates realistic 3D oceans with wave motion and wake waves, and accurate environment reflections. VRSG supports terrain built with bathymetry data for shoreline wave shape and opacity. Bathymetry data adds to the terrain higher shoreline fidelity and underwater geometry, making it ideal for terrain that will be used for littoral simulations. 3D wakes and ocean effects work the same with or without bathymetry data built into terrain tiles and in open ocean areas. In the current release, VRSG only supports 3D water at sea-level. VRSG will continue to render bodies of water which lie above sea level (rivers, lakes, and ponds) with its legacy 2D multi-textured, animated, normal-mapped water surfaces.

The terrain tiles will seamlessly match complete water tiles generated by VRSG, so you need not build terrain tiles that are fully water.



Views of the Canary Islands terrain shown in Terrain Tools (upper left, within ArcMap), and VRSG's real-time view of virtual Tenerife Island with 3D ocean simulation.

To learn more about the VRSG round-earth terrain format, see the chapter "VRSG Terrain Architecture."

Benefits of MVRsimulation's terrain tools

In addition to generating 3D terrain in round-earth VRSG terrain format, key benefits of MVRsimulation's Terrain Tools for ArcGIS include:

- Live compositing display of raster imagery and elevation (including bathymetry) data in a WYSIWYG interface.
- Support for any format of source data supported by ArcGIS.
- Raster display capabilities such as pan-sharpening, custom band order, multiple resampling techniques, histogram stretching, contrast and brightness control, masking, and edge blending.
- Ability to supply vector data to define linear and areal features. Generate road networks, fine tune elevation with polygon or point data, or create crisp boundaries between compiled terrain and VRSG-generated water representation.
- Ability to manage distributed terrain builds on a local network with web browser user interface. Optionally, use ArcGIS Engine for lower-cost headless build machines.
- Ingest external 3D content built upstream as part of the art pipeline.

All features integrate seamlessly with the industry's leading GIS platform.

Within the ArcGIS interface, coastlines and road networks can be digitized and cut out from geospecific imagery using Terrain Tools. At runtime, VRSG generates 3D oceans and multi-textured, animated, normal-mapped water surfaces in the regions identified as water, or blends road textures with underlying imagery. The terrain tiles seamlessly match complete water tiles generated by VRSG.

Once the 3D terrain has been built in Terrain Tools, you can use MVRsimulation's Scenario Editor to add culture and create and edit real-time pattern-of-life scenarios to play back in VRSG. Scenario Editor provides a graphical interface with tools and content libraries you can use to build dense 3D scenes with realistic visual characteristics and pattern-of-life scenarios. Experienced VRSG users and novices alike can work in a flexible manner to increase the realism of terrain easily with rich culture and scripted movements of vehicles and characters. The 3D terrain that you work with in Scenario Editor is the same 3D terrain you visualize in VRSG, and the scenarios you create can be run in both Scenario Editor and VRSG.

If you use VRSG with Battlespace Simulations' Modern Combat Environment (MACE), you can import the same road, building, or water vector source data compiled in terrain with Terrain Tools into the MACE mission on that terrain to ensure correlated features. You can also import cultural feature files exported from Terrain Tools files into MACE.

System requirements

The MVRsimulation suite of Terrain Tools for Esri ArcGIS requires:

- A PC system with an Intel Core i7 processor, running the Windows 10 64-bit operating system. The minimum required RAM is 8 GB; recommended RAM is 16 GB.
- Any license level of Esri ArcGIS Desktop version 10.4 and above (Basic, Standard, or Advanced). Additional terrain build machines can use either ArcGIS Desktop or ArcGIS Engine version 10.4 and above.

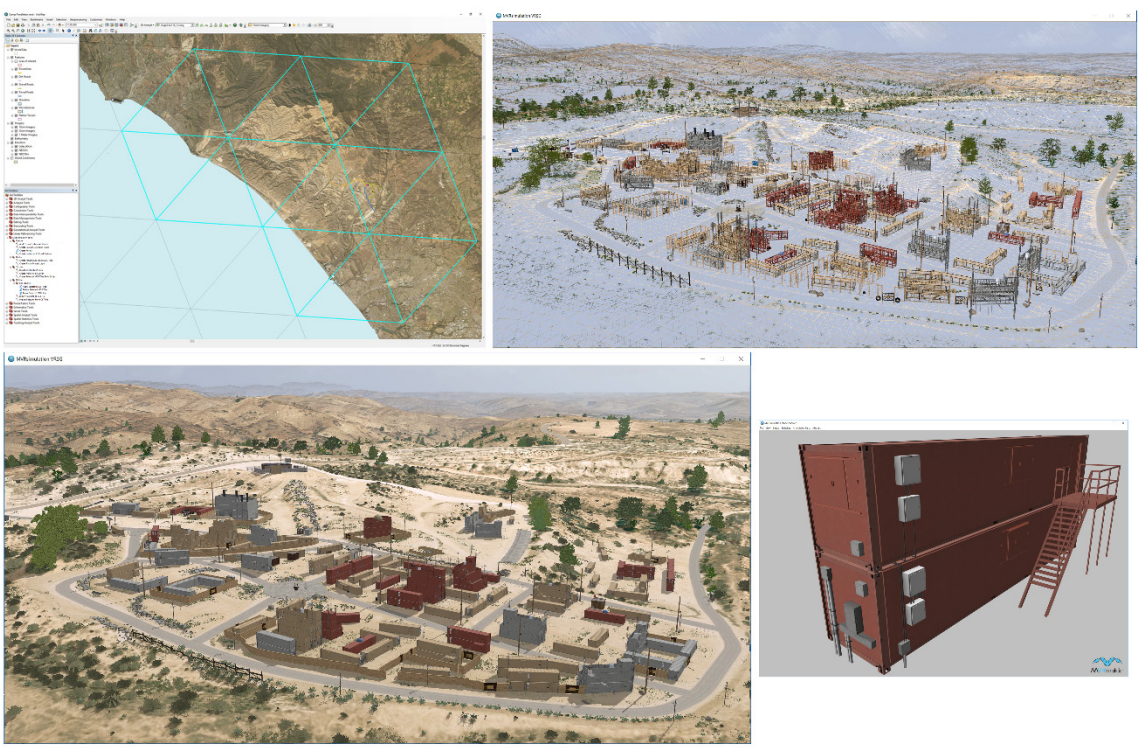
  Note: There is currently no roadmap for Terrain Tools to support ArcGIS Pro, as ArcGIS Pro uses a completely new developer SDK and does not contain the 3D APIs required for

Terrain Tools. ArcGIS Desktop continues to evolve and will continue to be supported by Esri for many years.

- Esri 3D Analyst Extension to ArcGIS.
- ArcGIS for Desktop Background Geoprocessing (64-bit) add-on software. ArcMap and ArcCatalog are 32-bit applications. The 64-bit version of ArcGIS background processing, which is delivered with ArcGIS as an optional installation, enables Terrain Tools to execute in an external 64-bit process for compiling 3D terrain. This add-on software *must be installed* prior to installing Terrain Tools. For more information about ArcGIS for Desktop Background Geoprocessing (64-bit) add-on, see http://desktop.arcgis.com/en/arcmap/latest/analyze/executing-tools/64bit-background.htm.
- Ample storage for compiling VRSG terrain tiles. When considering storage requirements, bear in mind that the terrain tiles do not need to reside on one machine. During terrain tile compilation, you can have newly built tiles output to multiple machines, each with a modest amount of storage.

The Esri products must be installed on your PC system before you install MVRsimulation Terrain Tools for ArcGIS.

The terrain that is generated by the MVRsimulation Terrain Tools requires the latest release of VRSG to take advantage of Terrain Tools' support of the latest features in VRSG.



Terrain Tools map view in ArcGIS of MVRsimulation's virtual replica of the Kilo 2 MOUT site at Camp Pendleton, a bird's-eye view photo of the MOUT site, a real-time bird's-eye view of the 3D terrain tiles rendered in VRSG, and a MOUT model displayed in Model Viewer. Photo (upper right): USMC.

What's new in this release

Terrain Tools for ArcGIS version 2.0 contains the following new features and enhancements:

- New Tile Set Layer feature, which provides the ability to use a one or more terrain tiles as imagery and/or elevation source. With this feature you can use existing tiles built by MVRsimulation as part of your site's terrain building effort, blended with your site's own imagery and elevation source data.
- Build Manager improvements:
  - Ability to rebuild one or more tiles from an existing build, optionally with different a minimum texture and/or elevation resolution.
  - Ability to inspect a tile in the Model Viewer without leaving Terrain Tools.
- CityEngine SDK integration, which provides compilation performance improvement with support for instancing and automatic LOD generation.
- New areal forest feature, which quickly populates a given polygon with points (with a buffer region around each) for dense tree coverage.
- Ability to generate 2048 x 2048 pageable textures automatically during compilation, which can be rendered in VRSG v7 and higher.
- The new name convention for cultural feature files and microtextures is vrsg.clt and vrsg_microtexture_ x_y_z.tex. Previously created cultural feature files and microtextures with the old convention of "metadesic" their name are respected by Terrain Tools and VRSG 7.

Automatic generation of LODs for runway features. With this release MVRsimulation is making available its library of City Engine rule packages (rpks), which were created in recent years for several terrain datasets. You can obtain these .rpks on MVRsimulation's Download Server, in /Software/Terrain Tools/. (Contact MVRsimulation at downloads@mvrsimulation.com if you need an account on the Download Server.)

All the advancements in the latest ArcGIS for Desktop version 10.8 are supported in Terrain Tools.

Previous releases

Version 1.6

- Ability to recompile an existing 3D terrain's cultural features (such as models, instances, lights, and fences) without having to recompile the terrain and textures.
- New feature type for compiling extruded building features using an Esri CityEngine rule package (.rpk) to enable adding large areas of dense culture to 3D terrain quickly and easily.
- Colorization of tree models and other vegetation by automatic color matching of the models with underlying geospecific imagery. The result yields increased realism and suspension of disbelief to flight simulations.
- The surface depth along a coastline is now computed as a function of the distance to the coastline. Bathymetry source data is no longer required to generate a bathymetry surface.

- Unique compiled-in models in MVRsimulation's HPX model format are no longer compiled as instanced geometry.
- Full support of all node types in MVRsimulation's HPX model format, such as external references and multi-texture geometry type.
- Powerline and light lobe support for compiled-in instances via positioning attributes specified in a JSON file.
- New tool to create a raster image from compiled terrain for cases where a heightmap is needed. Such cases include using the raster as input to CityEngine.
- Improvement to rendering transitions between levels of detail (LODs). This new technique smooths LOD transitions while reducing the triangle count by over 20%.
- Performance enhancements for large areas of instanced trees via LOD probability weights specified in a JSON file and support of the LOD_Scale feature attribute.
- Performance and navigation improvements to the Build Manager interface.
- Runway creation improvements to better support fitting multiple geometry layers within a runway model to the curvature of the earth and terrain profile.

Version 1.5

- MVRsimulation_Runway_Model polygon feature type for projecting a 2D runway model and cutting into the terrain.
- MVRsimulation_Terrain_Model polygon feature type for incorporating an external 3D terrain model by cutting out an area of terrain and replacing it with the model geometry during terrain compilation.
- Support for compiling large areas of instanced point features, such as trees, into the terrain.

Other MVRsimulation products

In addition to the Terrain Tools for ArcGIS, MVRsimulation develops VRSG, a 3D visualization product for visualizing geographically expansive and detailed virtual worlds on PCs. VRSG provides real-time single- or multiple-channel visualization of virtual environments, dynamic moving models, and special effects using Microsoft DirectX commercial standards.

High-resolution sub-inch per-pixel terrain imagery

MVRsimulation's small portable UAS (SUAS) collects high-resolution still-frame images at sub-inch (2 cm) per-pixel resolution, which after orthorectification can be used as source imagery to compile 3D terrain with MVRsimulation's Terrain Tools for Esri ArcGIS. The resulting geospecific synthetic environment can then be rendered in VRSG.

Aerial imagery collected by MVRsimulation's SUAS can be ordered directly from MVRsimulation by specifying the area of interest, such as an airfield. An aerial imagery order consists of:

- Raw collected photographic imagery data at sub-inch resolution.
- Orthorectified GEOTIFF imagery (processed by MVRsimulation) to be used with your Terrain Tools for ArcGIS software license.
- Dataset of terrain tiles in VRSG (round-earth) terrain format at 2 cm resolution to be used with your VRSG software license.

Real-time VRSG rendering of 2 cm terrain built with imagery collected by MVRsimulation's small UAS of the Naval Air Station Fallon Range Training Complex.

Real-time VRSG rendering in sensor view of the 2 cm terrain shown on the left. The higher the resolution of the visual database, the more accurate the IR profile for sensor simulation.

Aerial imagery orders must be for an area of a minimum of 20 sq km. Access to the area of interest for aerial photography *must* be in accordance with FAA regulations. Customers will be responsible for obtaining the necessary authorization and certified access to operate within the particular area of interest for aerial photography. Customers have unlimited unrestricted use rights to the imagery. MVRsimulation retains the right to use the collected data and resulting terrain tiles and to redistribute the terrain tiles (typically as part of a larger set of terrain tiles of a region).

Examples of terrain built from sub-inch resolution imagery can be seen in VRSG-recorded flyover videos, created with the H.264 plugin, which are located on MVRsimulation's website.

Complete systems

MVRsimulation can deliver its software on state-of-the-art personal computers. A complete VRSG system with MVRsimulation software and available terrain databases preinstalled, includes a Windows PC with the latest in high-end PC gaming components (using Maximum PC as a reference).

Complete systems also include the following third-party products, which MVRsimulation resells to provide a total virtual reality solution:

- The latest in PC-based real-time 3D graphics accelerators.
- The latest in DirectSound and DirectSound 3D-capable sound card with a high-end speaker and sub-woofer.
- A 6DOF controller: 3Dconnexion SpaceMouse Pro or SpaceMouse Compact.

Information about these products can be found on the MVRsimulation website at www.mvrsimulation.com.

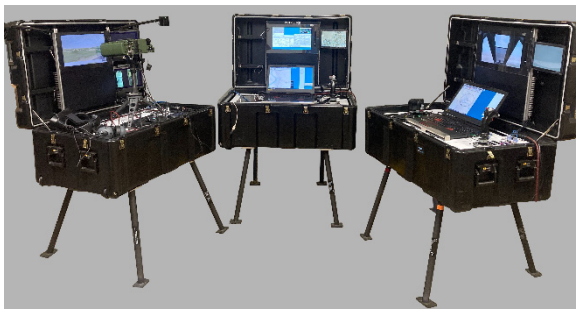
Software maintenance updates

A purchase of a new MVRsimulation Terrain Tools license is delivered with software maintenance for one full year. Maintenance includes technical support and software upgrades within the one-year period. If you are a current Terrain Tools customer, you can purchase a year of software maintenance directly from the order form on the MVRsimulation website, where complete pricing and order information is available.

Deployable Joint Fires Trainer

MVRsimulation's portable joint fires training solution, the Deployable Joint Fires Trainer (DJFT) provides a quick deploy capability for JTACs and forward observers to train alongside fixed- and rotary-wing aircrew within a fully immersive, joint training environment. The DJFT is designed to meet the JFS ESC MOA accreditation for types 1, 2, 3, day, night, and laser controls, and digitally aided close air support (DACAS).

At the DJFT Observer station with the Varjo XR-3 mixed-reality headset, and BSI's emulated SOFLAM and IZLID equipment.



The DJFT with VRSG, BSI MACE, and the Observer, Instructor, and Aircrew stations.

The modular plug-and-play system, designed by a former JTAC/forward observer, is comprised of three or more stations fully contained within two-person portable ruggedized cases. The DJFT contains all the hardware required to run dynamic, full-spectrum JTAC/joint fires training scenarios, including laptop, GPS receiver, mixed-reality HMD system, and communication systems. Scenarios are run on VRSG and BSI's MACE.

The DJFT is an MVRsimulation internally developed R&D system, not funded by a government program. A customer would need to seek formal accreditation for the system. For more information, see the MVRsimulation website at www.mvrsimulation.com.

Part Task Mission Trainer

MVRsimulation's new portable fixed-wing Part Task Mission Trainer (PTMT), designed and built under an internal development program, provides a low-cost, quick-deploy cockpit training solution to fill the gap in current in-use mission tactics training toolkits for military fixed-wing pilots. The system aims to maximize suspension of disbelief for trainee pilots as they practice mission tactics and coordination as part of joint training operations in networked environments. It can also operate as a standalone training solution.

Flying in the PTMT, with VRSG rendering the out-the window (OTW) view on the curved display.

The PTMT with VRSG's OTW view on the curved display and simulated sensor view on the cockpit control panel.

Made in the USA with a welded aluminum enclosure, the PTMT uses operational representative aircraft hardware to conduct air-to-air or air-to-ground training scenarios. The system can be configured for training for current 3rd and 4th generation combat aircraft currently used by NATO nations by easily changing the position of the specially-designed, patent pending, flight control stick between side-stick and center-stick positions.

Scenarios are run on VRSG and BSI's MACE. VRSG provides the real-time 3D out-the-window and sensor views. BSI's full suite of tools enables a multi-mission virtual role playing in the air-to-air arena, to include tactical displays that are integrated with HOTAS controls and emulate real world tactical systems. This coupling of MACE with VRSG provides the degree of immersion ideally suited to training from solo part-task mission objectives to large-scale, distributed live-virtual-constructive (LVC) rehearsal of major combat operations.

About MVRsimulation

MVRsimulation Inc., founded in 1997, is dedicated to providing image generator software that runs on commercial off-the-shelf personal computers. MVRsimulation software is designed around the Microsoft family of products to maximize both the affordability and accessibility of high-speed virtual world creation and visualization software.

MVRsimulation's goal is to provide affordable simulation solutions to users who need compact, portable, virtual reality software to produce imagery at high throughput rates on state-of-the art personal computers.

For more information about the company and its products, visit the MVRsimulation website at www.mvrsimulation.com.

If you have specific questions about installing or using the VRSG system, or about installing or using an MVRsimulation product, you can contact support services at:

Email: support@mvrsimulation.com
Phone: 617-739-2667 Fax: 617-249-0151

If you need technical assistance…

MVRsimulation is committed to providing you with a high quality product experience. If you encounter a problem with this product, first try to determine whether the problem stems from an underlying problem with your system, making sure that your machine meets the system requirements listed on the MVRsimulation website at www.mvrsimulation.com/products/vrsg/vrsgsystemrequirements.html.

Contacting MVRsimulation support

If you need further assistance, contact MVRsimulation via email at support@mvrsimulation.com with details about the sequence of actions that led to the problem and any resulting error messages. Attach any screenshots that help to illustrate the problem. In addition, supply the following information:

- The product version number
- The operating system and version number
- Any special hardware or software configuration that might affect the problem

Most email to MVRsimulation support services is answered within 24 hours, Monday-Friday 9:00 am to 5:00 pm US Eastern Time.

Returning a damaged dongle

If you have a damaged Terrain Tools dongle, MVRsimulation will replace it upon request, as described in the Return Merchandise Authorization (RMA) instructions on the website at: www.mvrsimulation.com/howtobuy/returns.html. To return a damaged dongle for replacement:

1. Notice the dongle ID number located on the sticker that is affixed to the dongle. This dongle ID number contains information about what the dongle is licensed for (product and product maintenance). You must supply this dongle ID number in any communication you have with MVRsimulation regarding your damaged dongle.
2. Email rma@mvrsimulation.com with a request to obtain an RMA number for the return of your damaged dongle.
3. Send to MVRsimulation the damaged dongle or at a minimum, the metallic unit (which houses the electronic circuit board and memory chips) along with the RMA number. MVRsimulation will *not* replace a dongle for which you only return the purple plastic holder without the corresponding metallic unit.

For more information about MVRsimulation and its products, visit the MVRsimulation website at: www.mvrsimulation.com.

About this manual

This manual describes how to install and use MVRsimulation Terrain Tools for ArcGIS to build 3D terrain for visualizing in MVRsimulation's image generator, VRSG. MVRsimulation values any feedback you have on this manual. Email any comments or suggestions to: support@mvrsimulation.com.

CHAPTER 2

Building 3D Terrain: Tutorial

This chapter is a step-by-step tutorial that shows you how to build 3D terrain of downtown Baltimore, MD, and the nearby BWI airport, with MVRsimulation Terrain Tools. The exercises in this chapter introduce several types of terrain features. All of the Baltimore imagery, elevation, and bathymetry source data, shapefiles, and textures needed to complete this tutorial are included in the Terrain Tools software installation.

The topics covered are:

- Setting up a map document, geodatabase, and raster catalogs for the source data.
- Creating raster mosaic layers and applying blend regions.
- Adding terrain features, such as roads, bodies of water, CityEngine buildings from a CityEngine rule package (RPK file), fences, trees, and microtextures.
- Compiling all the data into 3D terrain, *terrain tiles* in MVRsimulation's round-earth VRSG terrain format.

Each topic is described in full in later chapters of this user guide.

Note that the chapter "Visualizing 3D terrain in VRSG" briefly shows how to use VRSG, which is described fully in the *MVRsimulation VRSG User's Guide*.

This tutorial is suitable for any user that has familiarity with Esri ArcMap. Follow the order of the exercises outlined below in cumulative order to complete the terrain building tutorial.

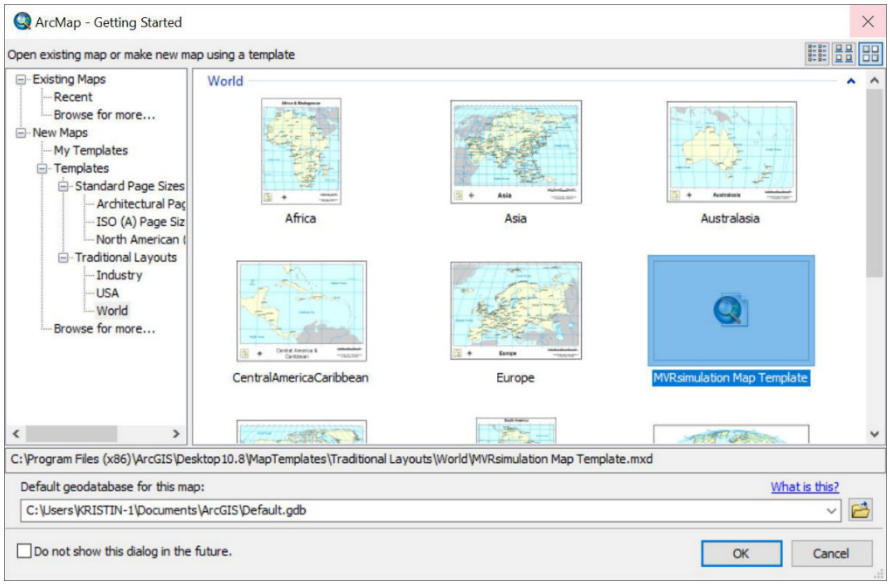
Setting up a new map document, geodatabase, and raster catalogs for data sources

MVRsimulation Terrain Tools uses the Esri ArcMap application as the user interface for building terrain. Launching ArcMap is the first step in starting a new terrain map document.

Setting up a new map document and associated geodatabase

To create a new map document (.mxd), using MVRsimulation's map template:

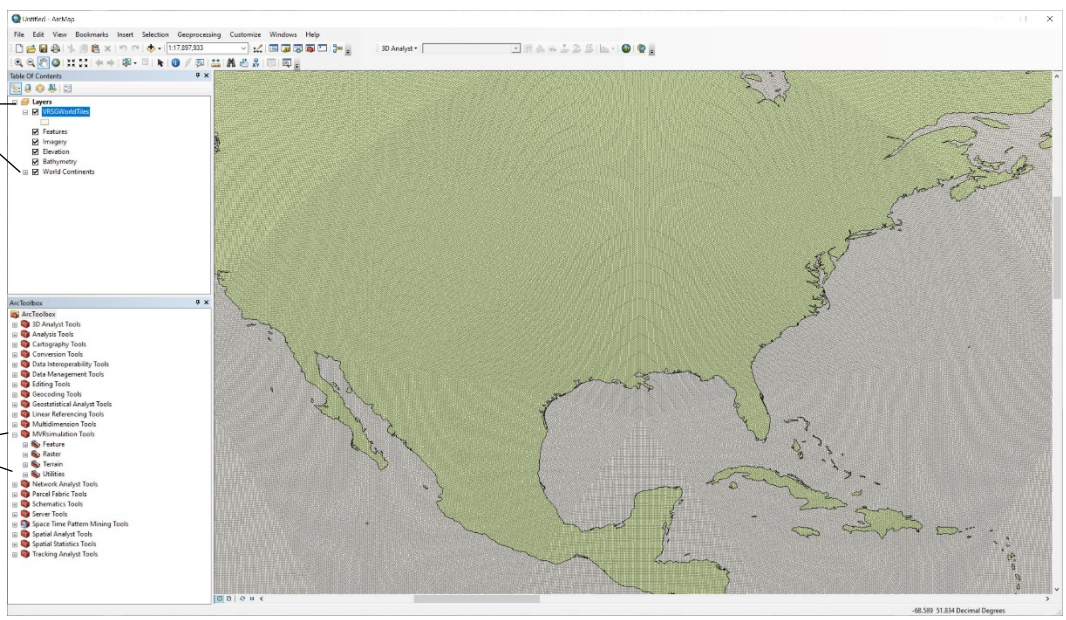
1. From the Windows Start menu, launch ArcMap. If this is the first time you have started ArcMap since the installation of MVRsimulation Terrain Tools, the MVRsimulation Software License Agreement will appear. Click OK to accept the license agreement.
2. In the ArcMap Getting Started dialog box, choose New Maps > Templates > Traditional Layouts > World on the left, and click on the MVRsimulation Map template. Click OK.



A map containing the outline of the world should be visible in the viewer as shown below:

Layers from the MVRsimulation map template listed on the ArcMap table of contents (TOC).

Four toolsets within the MVRsimulation toolbox. (If the MVRsimulation toolbox does not appear as shown, see the section "Accessing the MVRsimulation toolbox."

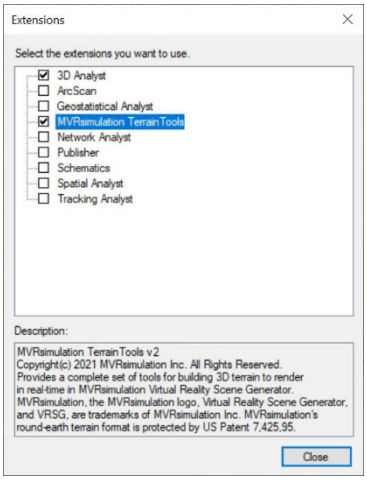


Before you go any further, take a moment to verify that:

- The MVRsimulation Terrain Tools hardware dongle is installed in your machine**.**
- The MVRsimulation Terrain Tools extension and 3D Analyst extension are enabled, as described next.

To check the status of the 3D Analyst and MVRsimulation Terrain Tools extensions:

1. Choose Customize > Extensions. When the Extensions dialog box appears, verify that the 3D Analyst and the MVRsimulation Terrain Tools extensions appear in the list and their corresponding checkboxes are selected. This indicates that the extensions are enabled.



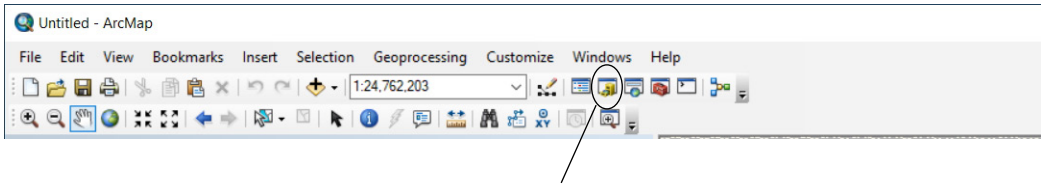
   If the MVRsimulation Terrain Tools extension is not enabled, click on its checkbox to display the Terrain Tools authentication dialog box. This dialog box provides instructions for obtaining and applying a license unlock code for the Terrain Tools product.

2. Click Close.
3. Choose File > Save to save your new map document in \MVRsimulation\ArcGIS Terrain Tools\Sample Project.
4. Name the map document (.mxd) Tutorial.mxd, and then click Save.

Adding connections to folders of source data

Before you begin, you'll need to connect to the Sample Project folder.

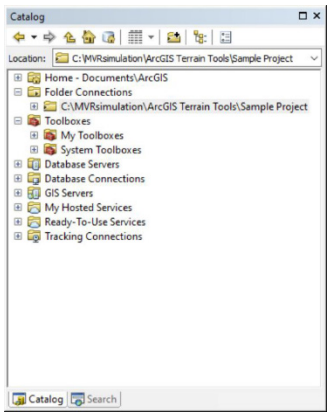
In ArcMap, launch ArcCatalog from the ArcCatalog tool on the ArcMap toolbar.



Shortcut on the toolbar for launching ArcCatalog within ArcMap.

The Catalog window enables you to access and manage your source data. To add folder connections, click the Connect to Folder button on the Catalog toolbar. To dock this window, click and hold on the top of the window, drag your cursor over to an arrow/chevron. The window will change size to show its docked position. Release and the catalog window should now be docked.

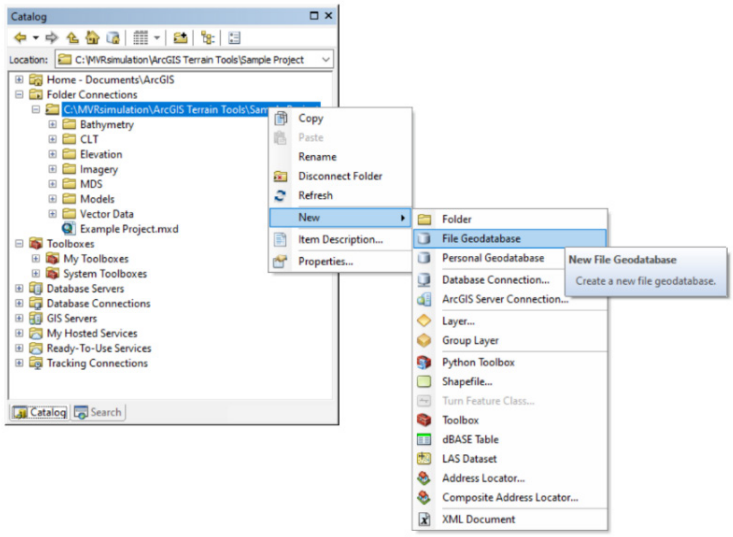
Map the folder location containing your source data to access and manage the data. For this tutorial, map the folder \MVRsimulation\ArcGIS Terrain Tools\Sample Project.



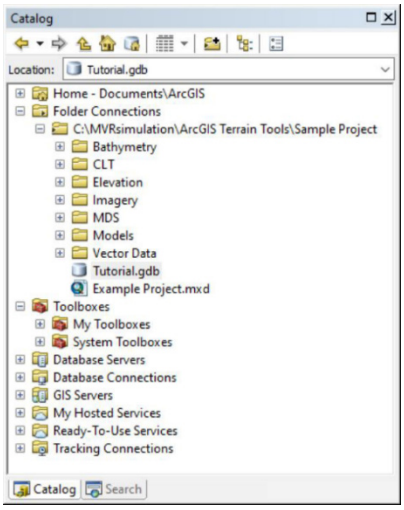
Creating a file geodatabase

A file geodatabase is a collection of source files in a folder on disk that can store, query, and manage geospatial data. You can create a file geodatabase from the Catalog tree or using the Create File GDB geoprocessing tool in the ArcToolbox.

1. In the Catalog window, right-click on the folder connection \MVRsimulation\ArcGIS Terrain Tools\Sample Project, and from the context menu choose New > File Geodatabase.



2. When the NewFileGeodatabase.gdb appears in your Sample Project workspace, right-click on it and choose Rename and rename it "Tutorial.gdb".

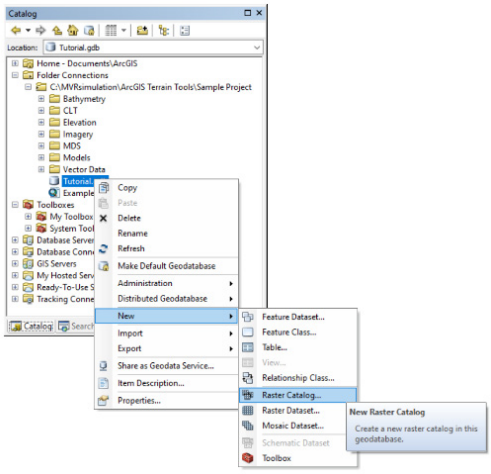


Creating raster catalogs to group data sources

A raster catalog is a container for managing raster datasets, such as imagery or elevation data and is stored within a file geodatabase. Raster catalogs are often used to display contiguous raster datasets. Raster data stored in a raster catalog must share a common coordinate system and resolution.

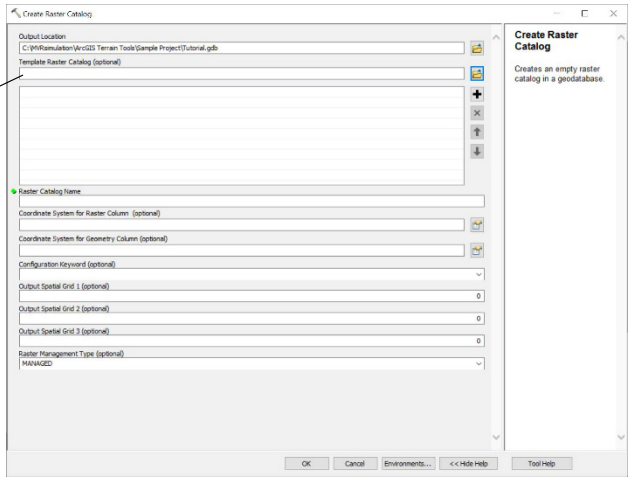
For the sample project, you will learn how to create multiple raster catalogs within a file geodatabase for managing different sets of raster data. (A raster catalog is also a prerequisite for the creation of MVRsimulation's Raster Mosaic Layer (RML), which you will create later in this tutorial.)

1. In the Catalog panel, right-click on Tutorial.gdb, and from the context menu choose New > Raster Catalog.

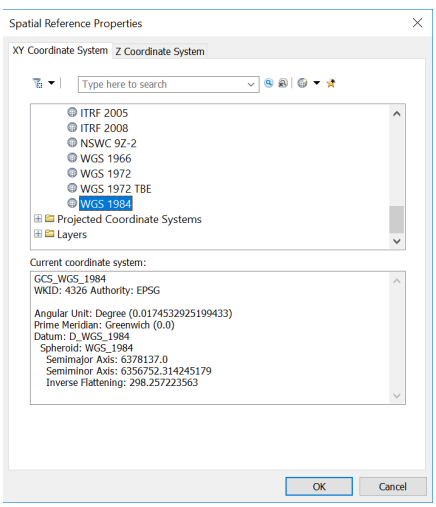


2. When the Create Raster Catalog dialog box appears, locate the Raster Catalog Name field and type the name "Baltimore_15cm_RC". Note that spaces or dashes are not allowed in names in ArcGIS.

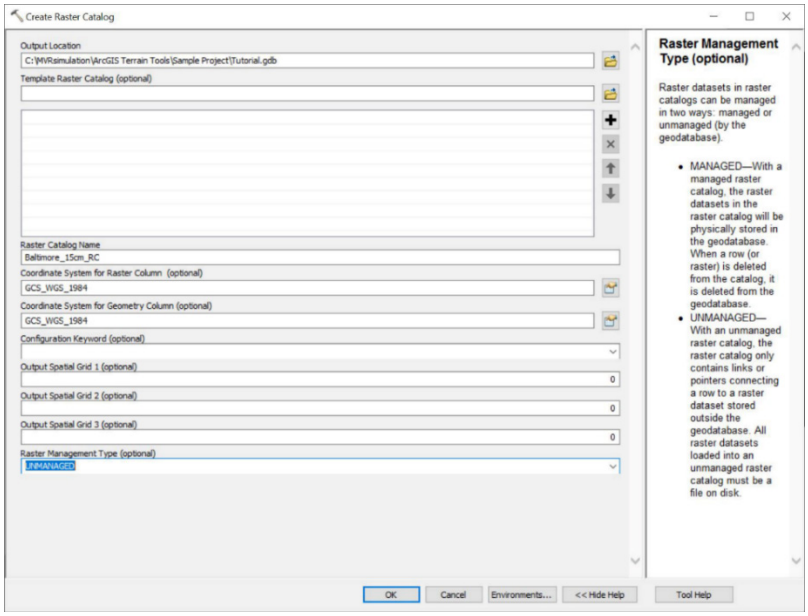
Type "Baltimore_15cm_RC" for the Raster Catalog name.



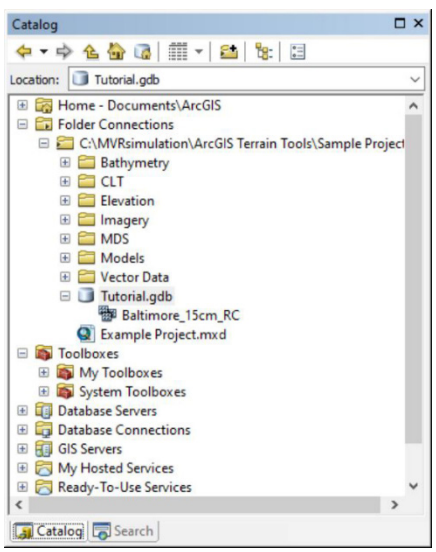
3. Click the Coordinate System for Raster Column browse button to specify a coordinate system for the raster catalog. Choose Geographic Coordinate System > World, choose WGS 1984, and click OK.



4. At the bottom of the Create Raster Catalog dialog box, select Unmanaged for Raster Management Type, and then click OK. (Unmanaged is the recommended option when using Terrain Tools.)



Notice the raster catalog now appears in both the catalog and the TOC.



Next, create a raster catalog for the KBWI_Baltimore_15cm imagery by following the same steps as above:

1. In the Catalog panel, right-click on Tutorial.gdb, and from the context menu choose New > Raster Catalog.
2. When the Create Raster Catalog dialog box appears, scroll down to the Raster Catalog Name field and type the following name: “KBWI_Baltimore_15cm_RC”.
3. Click the Coordinate System for Raster Column browse button and then choose Geographic Coordinate System > World and WGS 1984.

4. For Raster Management Type, select Unmanaged, and then click OK.

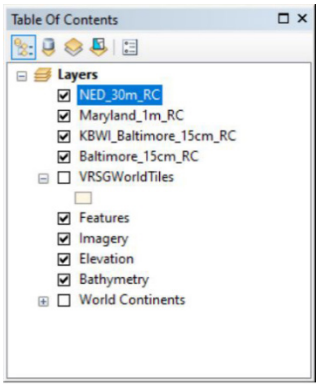
Finally, create a raster catalog for the Maryland_1m imagery by again following the same steps:

1. In the Catalog panel, right-click on Tutorial.gdb, and from the context menu choose New > Raster Catalog.
2. When the Create Raster Catalog dialog box appears, scroll down to the Raster Catalog Name field and type the following name: "Maryland_1m_RC".
3. Click the Coordinate System for Raster Column button, choose Geographic Coordinate System > World, and then choose WGS 1984.
4. For Raster Management Type, select Unmanaged, and then click OK.

Next, follow these same steps to create a raster catalog for the elevation data.

1. In the Catalog, right-click on your geodatabase Tutorial.gdb, and from the context menu select New > Raster Catalog.
2. For Raster Catalog Name, type the following in the text field: "NED_30m_RC".
3. Click the Coordinate System for Raster Column browse button and then select Geographic Coordinate System > World > WGS 1984.
4. For Raster Management Type, select Unmanaged.

On the left side of the map, notice the Table of Contents (TOC) panel. Listed at the top of the layers in the TOC, you should now see the four raster catalogs layers (RCs).



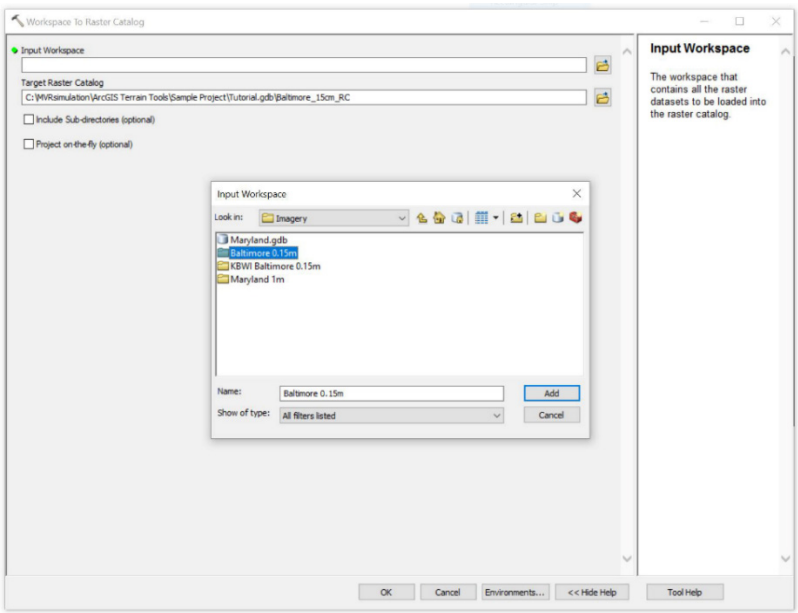
Loading raster data into the raster catalog

Once you have created the raster catalogs you will need to populate them with imagery and elevation data. Next, you will learn how to load all of the raster datasets stored in a common workspace into a raster catalog using Workspace to Raster Catalog.

In this section, you will load the sample project raster data into your raster catalogs.

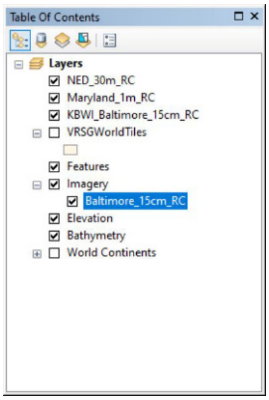
1. In the Catalog panel, right-click on the Baltimore_15cm_RC raster catalog, and from the context menu choose Load > Load From Workspace.

2. When the Workspace to Raster Catalog dialog box appears, click the Input Workspace browse button to navigate to the workspace from which you want to load the raster datasets. Navigate to \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Imagery and select the Baltimore 0.15m folder.



3. Select the Include Sub-Directories checkbox to load raster datasets from all the subfolders in your workspace, and then click OK to close the Workspace to Raster Catalog dialog box.

4. Click the Baltimore_15cm_RC layer in the TOC and drag the layer to position it under the Imagery layer.



5. Right-click on Baltimore_15cm_RC and select Zoom to Layer from the context menu.
6. You should now see in the map a wireframe representing the extents of the rasters within the Raster Catalog. For display performance efficiency, a wireframe view is displayed by default when a raster catalog (like this one) contains more than 9 rasters.
7. Zoom into the wireframe using the Zoom In tool from the ArcMap toolbar. Click and hold down the mouse to draw a rectangle to zoom into a square in the wireframe. You should now see a preview of the raster imagery in your map. It might take a minute to zoom in and redraw this raster catalog of high-resolution rasters.
8. To pan around the map, hold down the mouse wheel (or middle mouse button) or select the Pan tool from the ArcMap toolbar. Rolling the mouse wheel back and forth zooms in and out of the map view.
9. Now right click on Baltimore_15cm_RC and select Zoom to Layer from the context menu to get back to the full extent of the RC.
10. Save your map document by choosing File > Save.

Next, load the KBWI_Baltimore_15cm imagery into the KBWI_Baltimore_15cm_RC by following the same Load From Workspace steps 1 through 4 as described above. (Optionally, you can preview what the imagery looks like as described in steps 5-9 above,) Load the KBWI_Baltimore_15cm_RC with rasters from the following workspace location: \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Imagery\KBWI Baltimore 15cm.

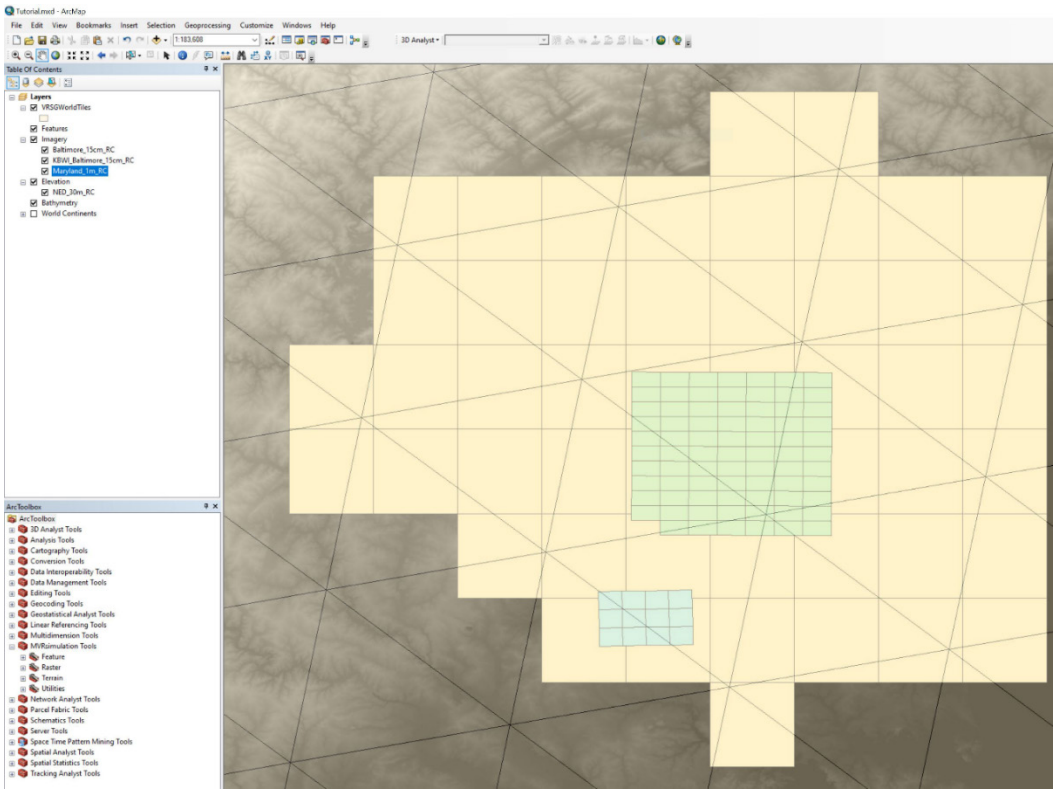
Next, load the Maryland_1m imagery into the Maryland_1_RC, again by following the same Load From Workspace steps 1 through 4 as described above. Load the Maryland_1m_RC with rasters from the following workspace location: \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Imagery\Maryland 1m.

Be sure all three imagery RC layers in the TOC are located under the Imagery layer.

Next, follow the same steps to load the Maryland elevation into the NED_30m_RC. The steps are below if you need to refer to them.

1. In the Catalog panel, right-click the NED_30m_RC raster catalog, and choose Load > Load From Workspace.
2. Click the Input Workspace browse button and navigate to the workspace from which you want to load the raster datasets. Navigate to \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Elevation\ and select NED 30m.
3. Select the Include Sub-Directories checkbox, and then click OK.
4. Drag the NED_30m_RC layer under the Elevation layer in the TOC.
5. Choose File > Save.

Your map document should look similar to the example below:



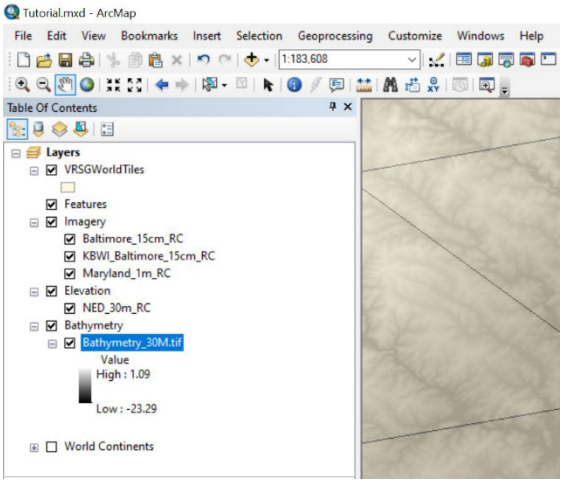
Notice that the elevation layer is displayed behind the imagery, as it is listed below the imagery in the TOC. Because there are many images to load, they are depicted in this view as polygons. You can, however, zoom in to inspect the imagery in detail.

Adding bathymetry data

The sample project has bathymetry data of the Chesapeake Bay, to include Baltimore's Inner Harbor area. This is elevation raster data of the shoreline and ocean floor. Bathymetry data may provide additional fidelity to the littoral areas.

To add bathymetry data:

1. In the Catalog panel, locate the Sample Project folder and then select and expand the Bathymetry folder (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Bathymetry).
2. Within the Bathymetry folder, select and expand the \Chesapeake Bay 30m folder.
3. Select and drag the Bathymetry_30M.tif from the Catalog panel to the TOC and place it under the Bathymetry layer as shown below:

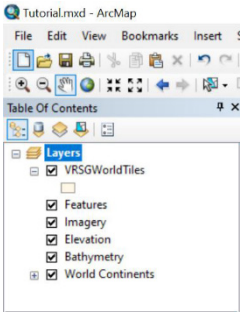


(If you are prompted to create pyramids for the Bathymetry raster data, click No. Disregard the ArcMap Geographic Coordinate System Warning; just click Close.)

About the layers in the tutorial map document

Source data is organized in the map template within the Table of Contents (TOC) layers.

The VRSGWorldTiles layer contains an outline of all of the possible VRSG terrain tiles you can select at build time. Each triangle represents the extents of a single terrain tile. If you zoom in closer you will see that the VRSGWorldTiles layer is made up of a dense mesh of triangles covering the entire Earth's surface. Each tile has a unique ID that also serves as the filename for the output MDS (.mds) file. These IDs are stored in the attribute table of the layer.



The Features layer is a group layer for all of your vector data including polygons, polylines, and point features. The order of these feature layers does not matter during the terrain compilation.

The Imagery layer is a group layer for storing all of your imagery raster data. The order of the raster layers should be organized from the highest resolution at the top, to the lowest resolution at the bottom. This layer is required for terrain compilation.

The Elevation layer is a group layer for storing all of your elevation raster data. You can also place 3D enabled vectors such as elevation flat patches here to smooth out the underlying

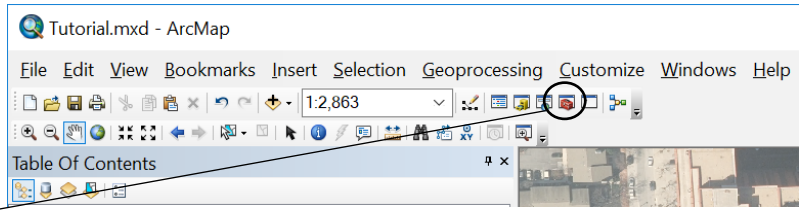
elevation data. The order of the raster layers should be organized from the highest resolution at the top, to the lowest resolution at the bottom. This layer is required for terrain compilation.

The Bathymetry layer is a group layer for storing bathymetric data. This layer is optional.

The World Continents layer is a standard map layer, purely for visual reference; it does not play a role in building terrain. You can unselect the checkbox for this layer if you do not need it for visual reference.

Accessing the MVRsimulation toolbox

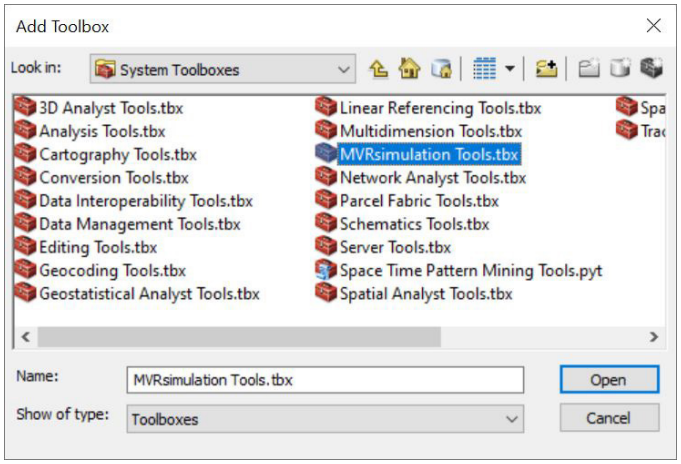
These next steps focus on adding terrain features and require the use of MVRsimulation Terrain Tools.



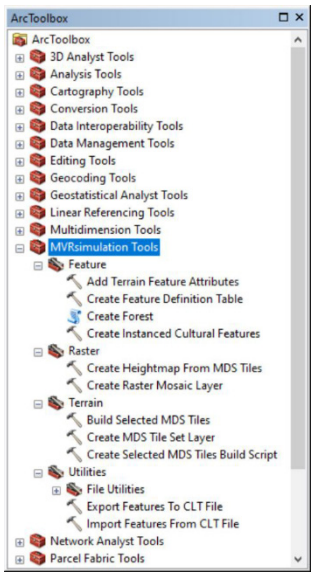
ArcToolbox button.

To add the MVRsimulation Tools toolbox:

1. Open ArcToolbox by clicking on the toolbox icon located on the file menu as shown in the image above. Optionally dock the window to a desired location.
2. Right-click on the red ArcToolbox button at the top of the ArcToolbox window and choose Add Toolbox.
3. When the Add Toolbox dialog box appears, choose System Toolboxes, choose MVRsimulation Tools.tbx from the System Toolboxes list, and then click Open.



Once you have located the MVRsimulation Tools toolbox, you can expand the toolbox to display the MVRsimulation toolsets within: Terrain, Features, Raster, and Utilities.



Working with imagery

In this section, you will create raster mosaic layers for the imagery and then apply a blending region for minimizing the discernable boundary where the high resolution (15 cm) Baltimore imagery meets the adjacent 1-meter Maryland imagery.

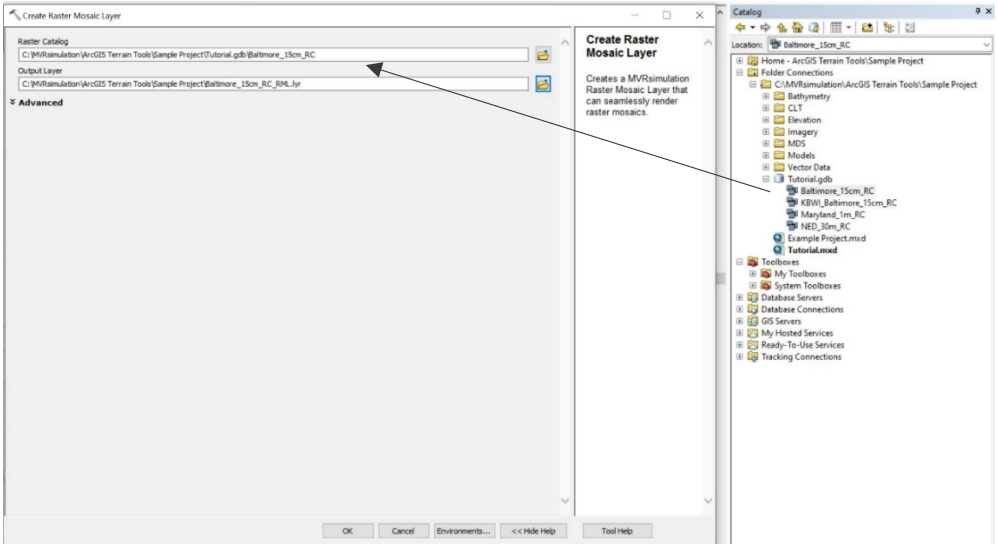
Creating raster mosaic layers

The Raster Mosaic Layer is a custom layer type in MVRsimulation Terrain Tools that helps with performing refinement tasks on rasters (such as adjusting color settings or edge blending) by acting on thousands of imagery rasters as a single imagery mosaic, and with processing geospatial imagery into terrain textures.

Note: The creation of a raster mosaic layer is typically not needed for elevation datasets.

You convert a Raster Catalog to a Raster Mosaic Layer with the Create Raster Mosaic Layer tool in the Raster toolset, as described in the steps in this section.

1. Open the ArcToolbox and expand MVRsimulation Tools. Next, expand the Raster toolbox and double-click on Create Raster Mosaic Layer.
2. When the Create Raster Mosaic Layer tool appears, click and drag 'Baltimore_15cm_RC' from the Catalog into the Raster Catalog field as shown below. (Alternatively, navigate to the location of the RC by clicking on the folder button and navigating to the location of the RC.)



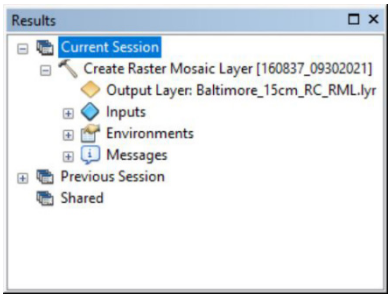
Notice the Output Layer field will automatically be populated with the corresponding name of the new raster mosaic layer created from the raster catalog.

3. Click OK. (This operation may take several minutes to complete.) Once complete, the raster mosaic layer will automatically appear listed at the top of the TOC.
4. In the TOC, right-click on Baltimore_15cm_RC_RML and select Zoom to Layer from the context menu. You should now see the full extents of your new mosiac. (Again, each zoom action on high-resolution imagery causes a redraw that will take a minute to execute.)
5. In the TOC, click and drag the Baltimore_15cm_RC_RML under the Imagery layer.
6. In the TOC, right-click on Baltimore_15cm_RC and select Remove. This deletes the Raster Catalog layer in the TOC; it is no longer needed now that the Raster Mosaic Layer for the 15 cm imagery has been created.
7. Choose File > Save.

Next, apply the steps above to create a Raster Mosaic Layer (RML) for the KBWI 15 cm imagery RC.

1. Using the MVRsimulation tool Create Raster Mosaic Layer, create an RML layer for the KBWI_Baltimore_15cm_RC raster catalog.
2. When the Create Raster Mosaic Layer tool appears, click and drag KBWI_Baltimore_15cm_RC from the Catalog into the Raster Catalog field.

   When Terrain Tools processes are submitted to ArcMap, messages about the executed tasks in the current session are captured in the Results panel on the left, like the following:



This display can help confirm success, warnings, or issues that might arise during terrain-building tasks.

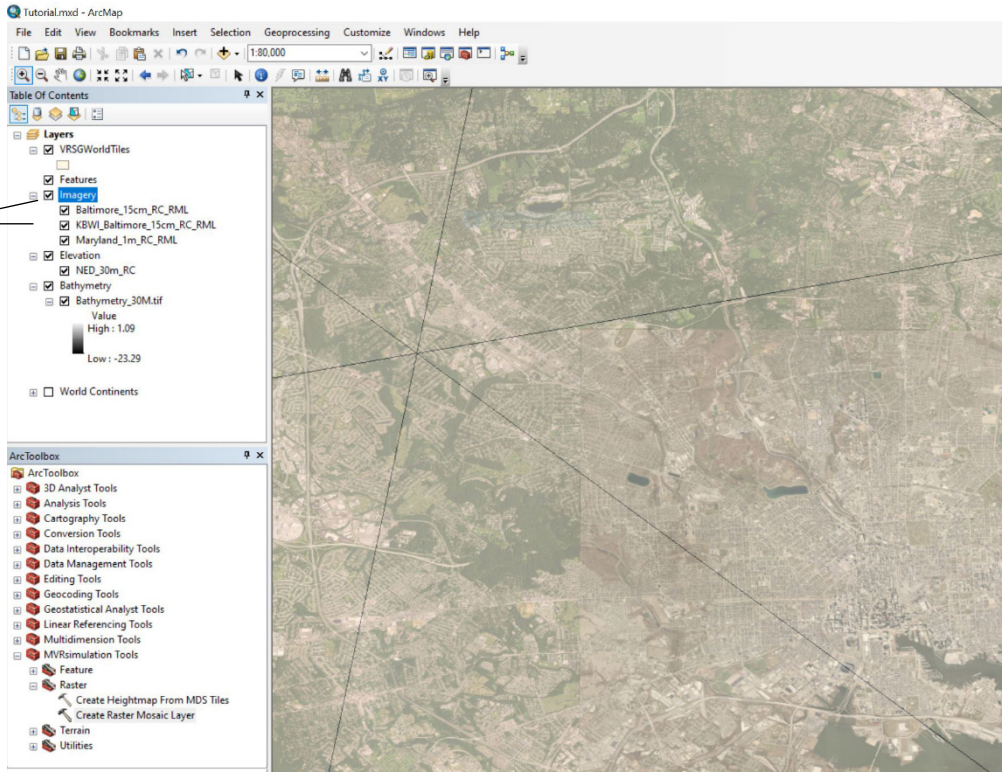
3. When the KBWI_Baltimore_15cm_RC_RML layer appears listed at the top of the TOC drag the KBWI_Baltimore_15cm_RC _RML layer to move it under the Baltimore_15cm_RC_RML layer.
4. In the TOC, remove the KBWI_Baltimore_15cm_RC layer now that you have added the RML layer for the airport's 15 cm imagery.
5. Choose File > Save.

Next, apply the steps above to create a Raster Mosaic Layer (RML) for the Maryland 1 meter imagery RC.

1. Using the MVRsimulation tool Create Raster Mosaic Layer, create an RML layer for the Maryland_1m_RC raster catalog.
2. When the Create Raster Mosaic Layer tool appears, click and drag Maryland_1m_RC from the Catalog into the Raster Catalog field and click OK.
3. When the Maryland_1m_RC_RML layer appears listed at the top of the TOC move the Maryland_1m_RC_RML layer to position it under the Baltimore_15cm_RC_RML layer.
4. In the TOC, remove the Maryland_1m_RC layer now that you have added the RML layer for the 1-meter imagery.
5. Choose File > Save.

Your map document should look similar to this next example:

List the imagery in the Imagery layer according to resolution, with the highest resolution imagery listed first.



Again, you should list the highest resolution imagery above any lower resolution imagery in the Imagery layer of the TOC. This order affects terrain compilation.

Note: Each time you zoom in the map document (either by choosing Zoom to Layer or using the Zoom tool), it will take a few moments for the map to refresh as the imagery is redrawn. The main cause of the slow refresh is the redraw of the high-resolution Baltimore_15cm_RC_RML. To speed up the zoom actions while you work on the tutorial steps, you can turn off the display of this layer temporarily. But you *must turn on* the display of this layer before you build terrain tiles.

To turn off the display of this or any layer, in the TOC, unselect the checkbox to the left of the layer. To turn on the layer display, select the checkbox.

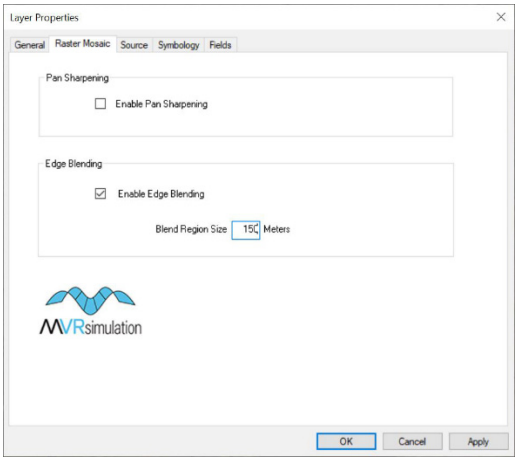
Applying a blending region to imagery boundaries

When using imagery of different resolutions, typically a noticeable change in detail and sometimes in color occurs at the boundary where the higher resolution imagery meets the adjacent lower resolution imagery.

In this section, you will minimize the discernable boundary where the high resolution (15 cm) Baltimore imagery meets the adjacent 1-meter Maryland imagery by applying feather-blending to a region to soften the transition between the adjacent imagery.

1. In the TOC, right-click on the Baltimore_15_cm_RC_RML layer and select Zoom to Layer.
2. Using the Zoom In tool located on the ArcMap toolbar, zoom in further on an area where you can see where the edge of the15 cm imagery meets the adjacent 1m imagery. (It will take a few moments to redraw the imagery each time you click the Zoom tool.) An example of where the edges of two imagery resolutions meet is shown in the example below:

3. On the TOC, right-click on Baltimore_15cm_RC_RML and select Properties.
4. When the Layer Properties dialog box appears, click the Raster Mosaic tab.
5. Select the Edge Blending checkbox.



6. In the Blend Region Size field, type “150” meters, click Apply, and then click OK.
7. In the TOC, right-click on Maryland_1m_RC_RML and select Zoom to Layer so you can examine the blending improvement in the imagery transition.

See the chapter “Refining Terrain Construction” for more information about color correction and adjustments you can make to a Raster Mosaic Layer to improve the visual quality of the resulting 3D terrain.

At this point, you could build 3D terrain from these layers of imagery and elevation data. Instead, in the next section, you will add some cultural features to enhance the realism of the terrain.

Before going further, save your map document by choosing File > Save.

Adding terrain features

Terrain features in Terrain Tools are polyline or polygon feature types for cultural features added to the terrain, such as buildings, trees, roads, bodies of water, and so on.

- Polyline terrain feature types are roads, walls and fences, and cultural lights.
- Polygon terrain feature types are bodies of water, buildings, 2D runways, and 3D inset terrain models, and microtextures.

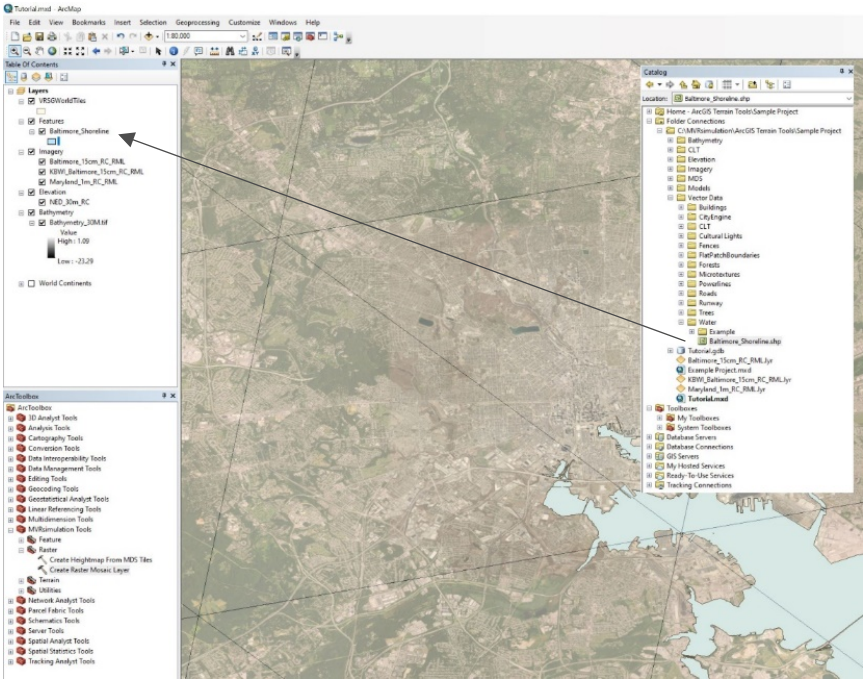
This section contains steps for adding shapefiles for each feature type. The tutorial shapefiles are located in \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data.

For further information about feature processing beyond this tutorial, see the chapter "Working with Terrain Features." To learn how to make shapefiles, see the ArcMap help.

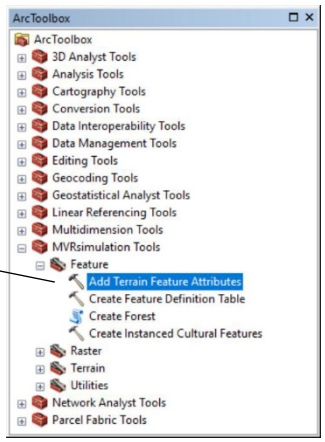
Adding a 3D water surface feature layer

The tutorial map document of Baltimore contains the Inner Harbor area on the Chesapeake Bay. With Terrain Tools, you can add a 3D water polygon feature to the terrain that will mark the harbor area covered by 3D ocean. When you build the 3D terrain, the appropriate areas will be marked as a 3D water surface in the resulting terrain tiles. When the resulting terrain is visualized in VRSG with 3D ocean simulation turned on, VRSG will render these water areas as 3D ocean, which results in much greater visual fidelity than using the overhead imagery to depict the harbor.

1. In the Catalog panel, locate the Sample Project folder and then select and expand the Vector Data folder (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data).
2. Within the Vector Data folder, select and expand the \Water folder.
3. Select and drag the Baltimore_Shoreline.shp to the TOC and place it under the Features layer as shown in the example below.

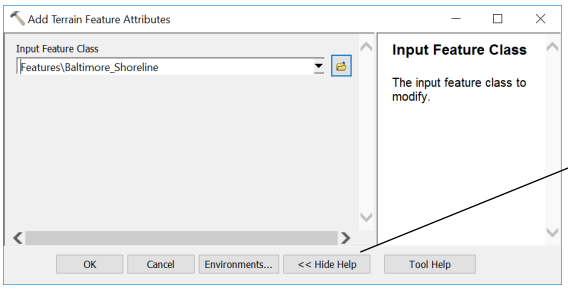


4. In the TOC, right-click on the Baltimore_Shoreline layer and select Zoom to Layer so you can better see the harbor area.
5. In the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on the tool Add Terrain Feature Attributes.



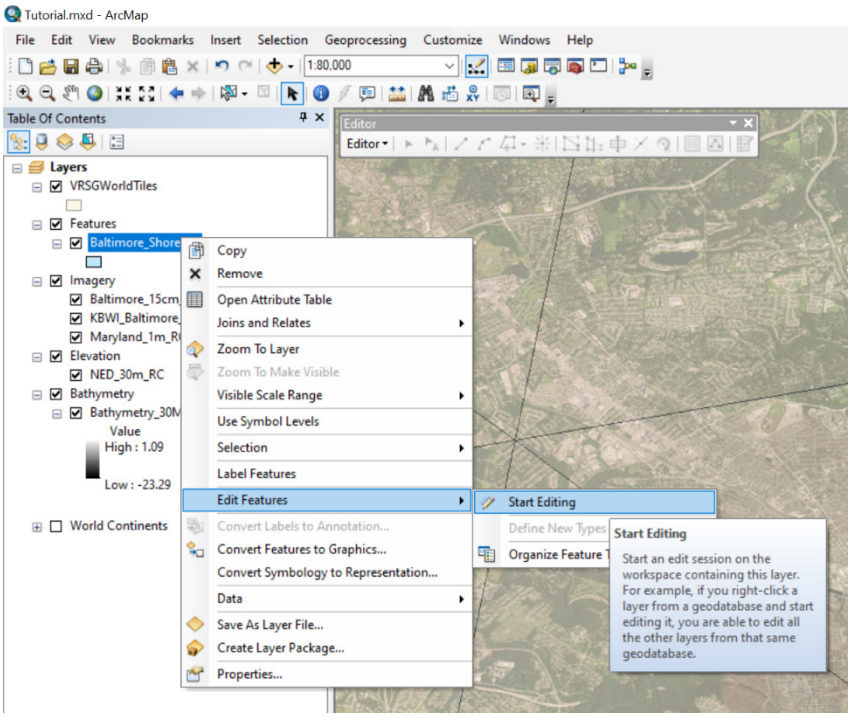
MVRsimulation tool for adding terrain features.

6. In the Add Terrain Features Attribute dialog box, click the down-arrow in the Input Feature Class field, select Baltimore_Shoreline from the list, and then click OK.

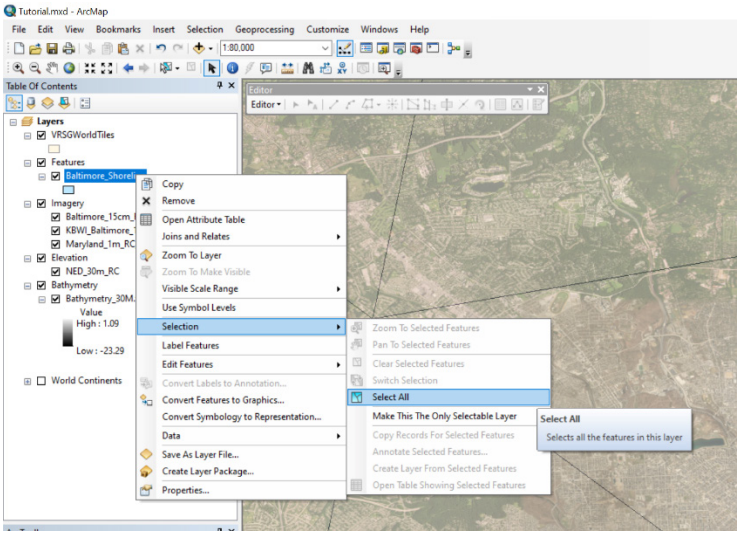


Click this button to show/hide the Help panel on the right side of the tool dialog box.

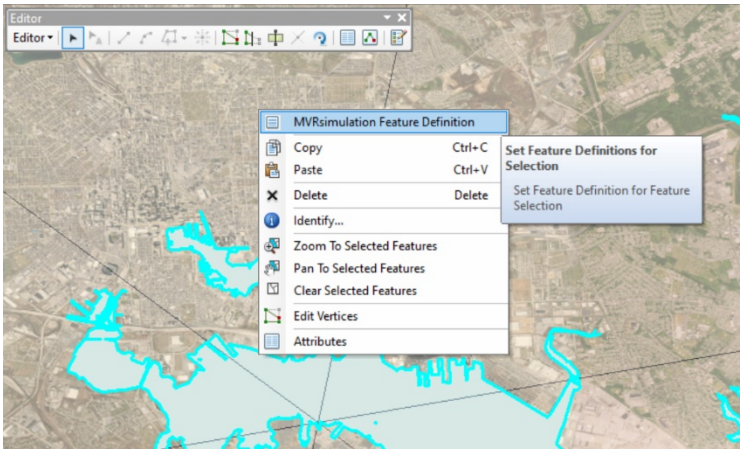
7. In the TOC, right-click on the Baltimore_Shoreline layer and choose Open Attribute Table. When the table appears, verify that a FeatureDef attribute column has been added to the table. Close the table.
8. In the TOC, right-click on Baltimore_Shoreline and from the context menu choose Edit Features > Start Editing.



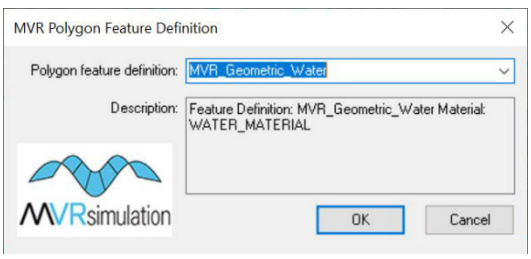
9. Ensure nothing in the map is selected, right-click on the Baltimore_Shoreline layer, and then choose Selection > Select All.



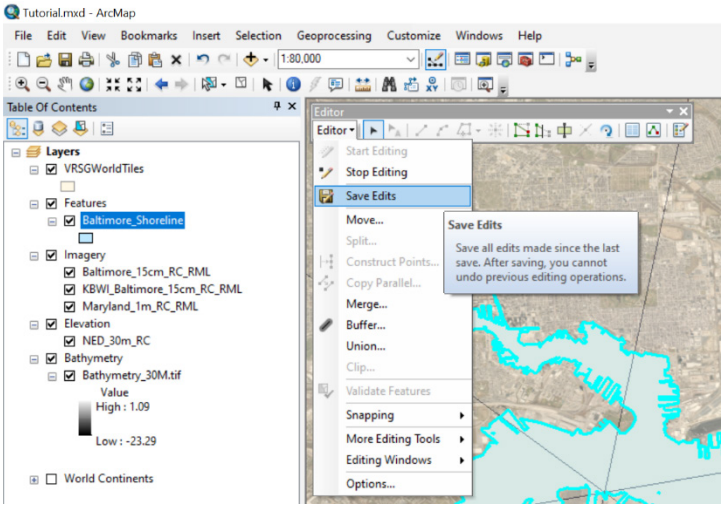
10. With the features selected, right-click anywhere in the map viewer and from the context menu select MVRsimulation Feature Definition.



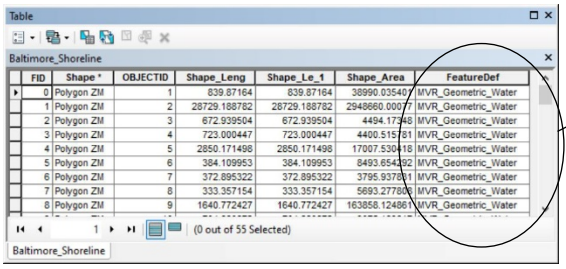
11. When the MVR Polygon Feature Definition dialog box appears, click the down-arrow in the Polygon Feature Definition field, select MVR Geometric Water, and then click OK.



12. From the floating Editor menu, choose Save Edits, and then choose Stop Editing.



13. Optionally, in the TOC, right-click on the Baltimore_Shoreline layer and select Open Attribute Table. When the table appears, verify that the FeatureDef attribute column has been populated with MVR_Geometric_Water.



Populated FeatureDef column.

Note: This optional step is merely to show the underlying ArcMap feature definition table that Terrain Tools is writing to.

14. Close the table and choose File > Save to save your work.

Terrain Tools offers two other polygon feature types:

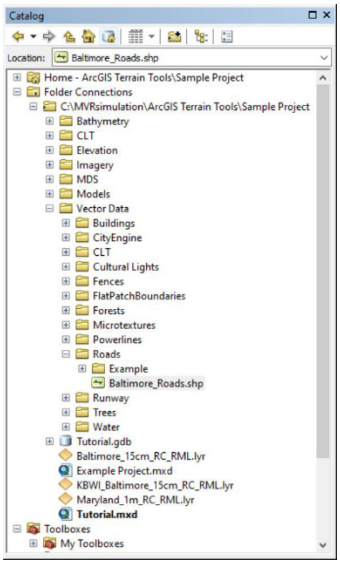
- CityEngine Buildings – For generating buildings using a CityEngine rule package (RPK). You don't need to have CityEngine to use an existing RPK file.
- Microtexture - For adding geotypical ground textures (*microtextures*) to one or more specific areas on the terrain to blend with the geospecific terrain imagery. When the terrain is rendered in VRSG, this texture adds further detail to the terrain when the eyepoint is close to the ground.

These feature types, which are included in exercises later in this tutorial and described in full in the chapter "Working with Terrain Features," can be added to the map in the same way as the water feature type, with an additional step for supplying values for some feature-specific attributes. The \Sample Projects directory contains shapefiles of these feature types so you can try them out. Textures to use with polygon feature types are located in the directory \MVRsimulation\ArcGIS Terrain Tools\Textures.

Adding roads

The \Sample Project directory contains a shapefile of downtown Baltimore roads that you can add to the tutorial map document as a polyline feature with an assigned road texture. For simulations that require participants to view the terrain at close range, having cut-out roadways textured with a high-resolution road texture adds to the realism of the virtual world. This has the additional benefit of flattening the road slope while smoothing out any artifacts in the elevation source data.

1. In the Catalog panel, locate the Sample Project folder and then select and expand the Vector Data folder (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data).
2. Within the Vector Data folder, select and expand the Roads folder and locate the shapefile Baltimore_Roads.shp.

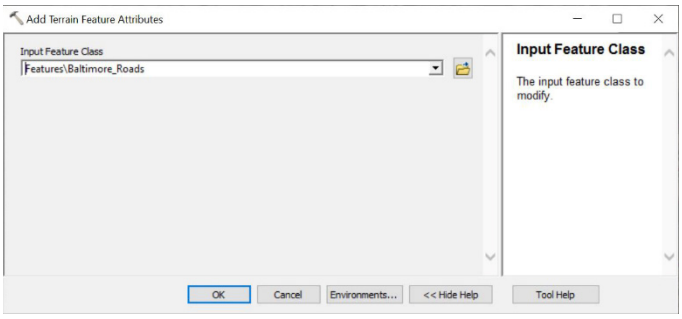


3. Drag Baltimore_Roads.shp from the Catalog to the TOC and place it under the Features layer.

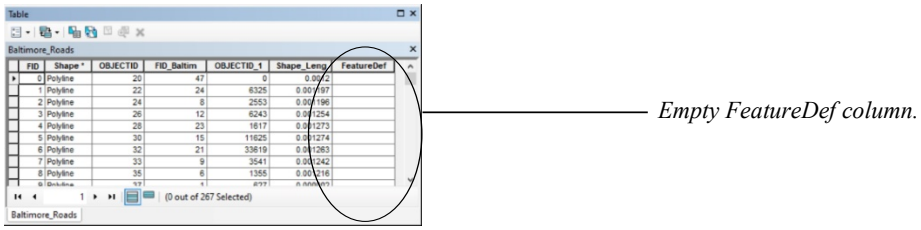
   In the TOC, right-click on the Baltimore_Roads layer and select Zoom to Layer so you can see the extents of the shapefile. You might need to zoom in further (using the Zoom tool on the ArcMap toolbar) to see the roads in the map.

4. In the MVRsimulation Tools toolbox, expand the MVRsimulation Terrain toolset and double-click on the tool Add Terrain Feature Attributes. This tool adds the Feature Definition field to the shapefile.

5. When the Add Feature Attributes dialog box appears, click on the down-arrow in the Input Feature Class field, select Baltimore_Roads from the list, and then click OK.

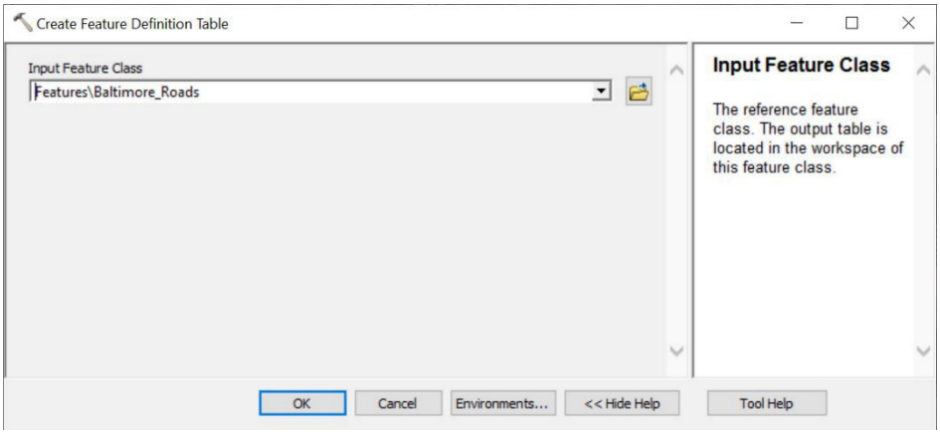


6. Optionally, open the attribute table to see the currently empty FeatureDef column.



A few more steps are needed to set up how the road should be textured during terrain processing. First, you need to create a feature definition table to store the feature attributes.

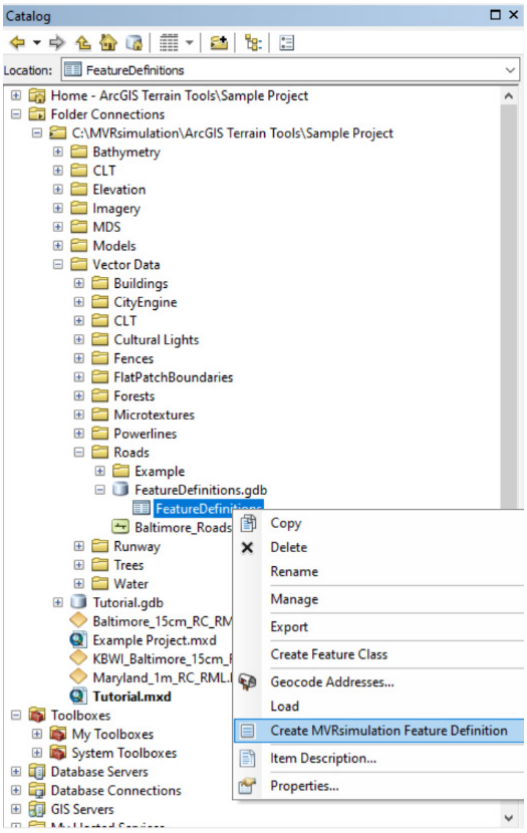
1. In the MVRsimulation Tools toolbox, expand the Feature toolset and double-click the tool Create Feature Definition Table.
2. When the Create Feature Definition Table dialog box appears, click on the down-arrow in the Input Feature Class field, select Baltimore_Roads from the Features list, and then click OK. This action creates an empty table with the required feature attribute fields and adds it to the map.



Next, you need to populate the Feature Definition Table to assign road attributes. Terrain Tools provides a menu for adding attribute values to the table, so you do not have to add them directly to the table.

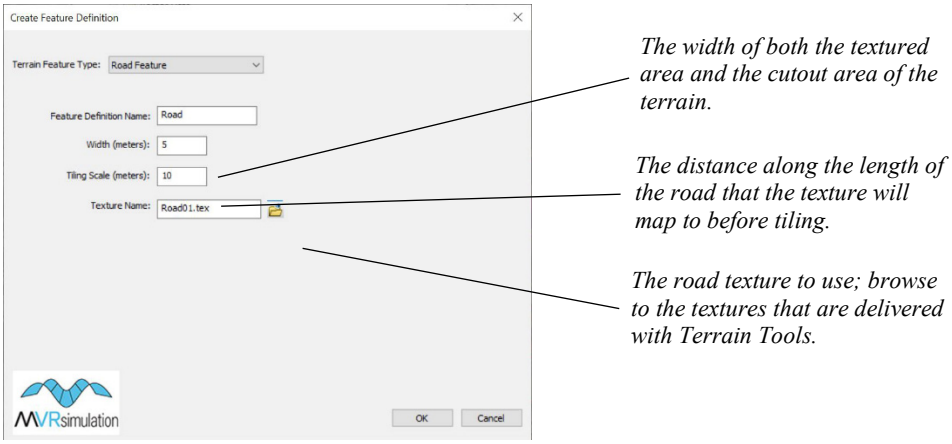
1. In the Catalog window, click on the Roads directory in your sample project which is located in MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\Roads. Right-click on this directory and select Refresh from the menu. You should now see a FeatureDefinitions.gdb.
2. Click on the + icon to expand the contents of the FeatureDefinitions.gdb. The feature definition table for your roads feature is stored in this file geodatabase.

3. Right click on the Feature Definitions table, select Create MVRsimulation Feature Definition.



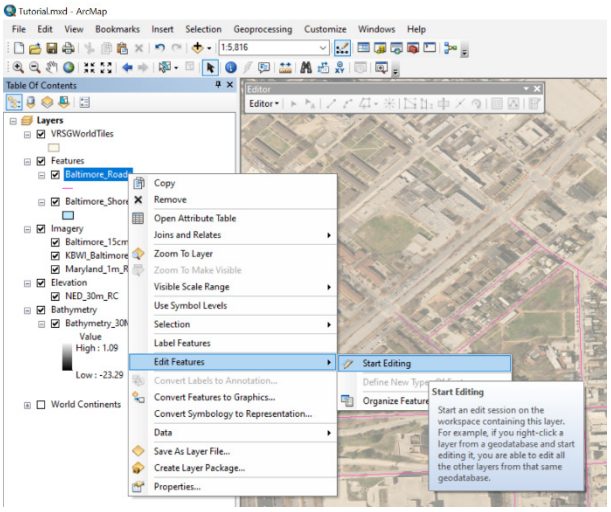
4. When the Create Feature Definition dialog box appears, fill in the following information:

   - For Terrain Feature Type, click the down-arrow and select Road Feature from the list.
   - For Feature Definition Name, type “Road”. This is a unique name that the feature uses to reference the feature definition.
   - For Width type “15”. This value specifies the width of the roads.
   - For Tiling Scale type “10”. This value is the distance before the texture repeats.
   - For Texture Name, click on the folder browse button and browse to the texture file Road01.tex located in \MVRsimulation\ArcGIS Terrain Tools\Textures\Road\. This is the texture file that will be used to texture the road during terrain processing.



5. Click OK.

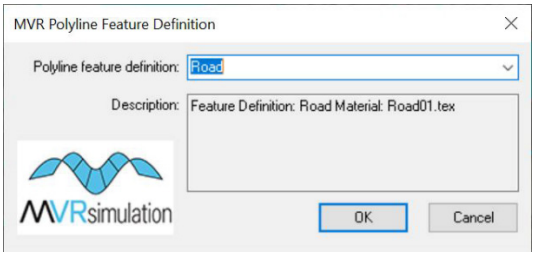
6. Next, in the TOC, right-click on Baltimore_Roads and from the menu choose Edit Features > Start Editing.



7. Ensure that nothing is selected in the map. Right-click on the Baltimore_Roads layer and select Selection > Select All.

8. With the features selected, right-click anywhere in the map viewer and from the context menu select MVRsimulation Feature Definition.

9. When the MVR Polyline Feature Definition appears, click the down-arrow on the context menu and select Road to assign your road feature definition to your selected road vectors. Click OK.



10. From the Editor menu, choose Save Edits, and then choose Stop Editing.
11. Choose File > Save to save your work.

Terrain Tools provides three other feature types that are covered in this tutorial:

- Forest Feature – for populating a large number of tree models into the terrain tiles.
- City Engine Feature – for compiling buildings into the terrain using a CityEngine RPK file.
- Wall Feature − for creating extruded 2D walls and fences.
- Cultural lights Feature − for creating a network of light points.
- Microtexture Feature − for assigning high resolution ground textures to the terrain.

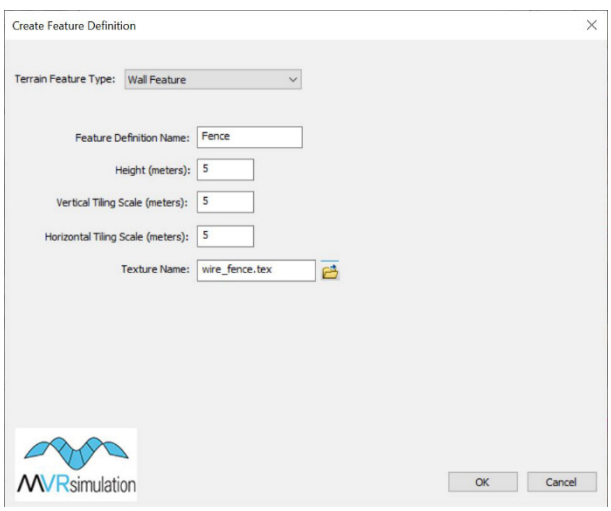
The Sample Project contains shapefiles of these feature types so you can try them out in the next section of the tutorial.

Adding extruded fences

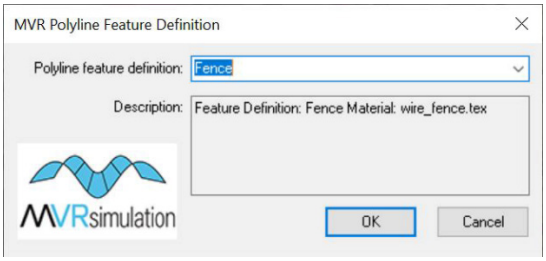
Preparing data for fence extrusion during terrain compilation is similar to the previous steps of preparing a road network. In this section, you will create a feature definition for fence creation.

1. In the Catalog panel, locate the Sample Project folder, and then select and expand the Vector Data folder (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data).
2. Within the Vector Data folder, select and expand the \Fences folder and locate the shapefile Baltimore_Fences.shp.
3. Drag Baltimore_Fences.shp from the Catalog to the TOC and place it under the Features layer. (The order of layers listed in the Features layers is not important.)
4. *(Optional)* In the TOC, right-click on the Baltimore_Fences layer and select Zoom to Layer so you can see where the fences are located.
5. In the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on Add Terrain Feature Attributes to assign a FeatureDef attribute to the attribute table.
6. When the Add Terrain Feature Attributes dialog box appears, click the down-arrow in the Input Feature Classes field, select Baltimore_Fences, and then click OK.
7. In the MVRsimulation toolbox, expand the Feature toolset and double-click on the tool Create Feature Definition Table.

8. When the Create Feature Definition Table dialog appears, click the down-arrow in the Input Feature Class field, select Baltimore_Fences from the Features list, and then click OK.
9. In the Catalog window, click on the Fences directory in your sample project which is located in MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\Fences. Right-click on this directory and select Refresh from the menu. You should now see a FeatureDefinitions.gdb.
10. Click on the + icon to expand the contents of the FeatureDefinitions.gdb.
11. Right click on the Feature Definitions table, select Create MVRsimulation Feature Definition from the context menu.
12. When the Create Feature Definition dialog box appears, enter the following information:
    - For Terrain Feature Type, click the down-arrow and select Wall Feature from the list.
    - For Feature Definition Name, type "Fence".
    - For Texture Name, click on the folder browse button, navigate to \MVRsimulation\ArcGIS Terrain Tools\Textures\Wall, and select wire_fence.tex. The rest of the fields can be left to their default values.



13. Click OK.
14. In the TOC, right-click on Baltimore_Fences and from the context menu select Edit Features > Start Editing.
15. Ensure nothing in the map is selected, right-click on the fences feature layer, and then click Selection > Select All.
16. With the features selected, right-click anywhere in the map viewer and from the context menu select MVRsimulation Feature Definition.
17. When the MVR Polyline Feature Definition dialog box appears, click the down-arrow, select Fence to assign the fence feature definition to the selected polylines. Click OK.



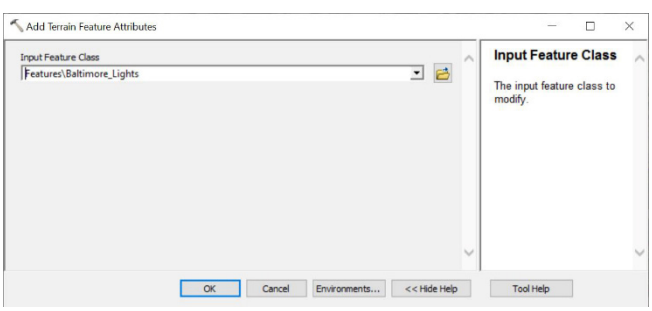
18. From the Editor menu, choose Save Edits, and then choose Stop Editing.
19. Choose File > Save to save your work.

Adding cultural lights

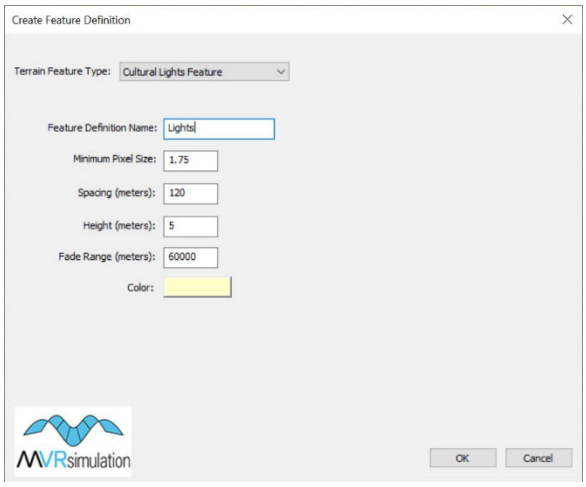
Adding cultural lights to the map document is similar to the steps you have completed for adding other polyline feature types (roads and fences). But for the Cultural Lights feature type you assign values to attributes for the color, size, and spacing of the lights.

For the purpose of this example, a cultural lights shapefile is provided with the Sample Project, in \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\Cultural Lights\Baltimore_Lights.shp. This shapefile contains a road network of the downtown Baltimore area.

1. In the Catalog window, locate the Sample Project folder and then select and expand the Vector Data folder (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data).
2. Within the Vector Data folder, select and expand the \Cultural_Lights folder, and locate the shapefile Baltimore_Lights.shp.
3. Drag Baltimore_Lights.shp from the Catalog to the TOC and place it under the Features layer. Notice the map is now filled with colored lines depicting where the lights will be when the terrain is built.
4. In the TOC, right-click on the Baltimore_Lights layer and select Zoom to Layer to see where the lights are located.
5. From the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on Add Terrain Feature Attributes to create a FeatureDef attribute in the attribute table.
6. When the Add Terrain Feature Attributes dialog box appears, click the down-arrow in the Input Feature Classes field, select Baltimore_Lights, and then click OK.

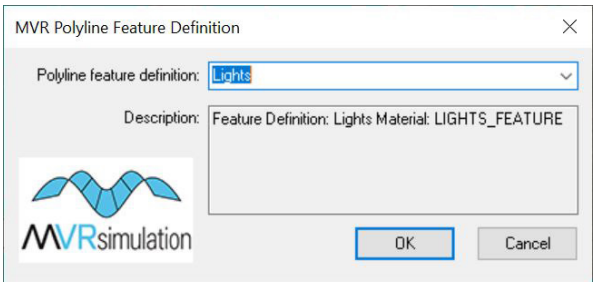


7. In the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on the tool Create Feature Definition Table.
8. When the Create Feature Definition Table dialog box appears, in on the down-arrow in the Input Feature Class field, select Baltimore_Lights, and then click OK.
9. In the Catalog window, click on the Cultural Lights directory in your sample project which is located in MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\Cultural Lights. Right-click on this directory and select Refresh from the menu. You should now see a FeatureDefinitions.gdb.
10. Click on the + icon to expand the contents of the FeatureDefinitions.gdb.
11. Right click on the Feature Definitions table, select Create MVRsimulation Feature Definition from the context menu.
12. When the Create Feature Definition dialog box appears, fill in the following information:
    - For Terrain Feature Type, select the Cultural Lights Feature.
    - For Feature Definition Name, type "Lights".
    - For this tutorial, you can leave the default values in place for the other attributes. Notice you can specify the pixel size of the lights, spacing between lights, and the height of each light. Also, you specify the fade range, which is the level-of-detail switch-out range at which the lights disappear in the distance. Clicking the color patch displays the Color Selector palette where you can specify the RGB values for the color of the lights.



13. Click OK.

14. In the TOC, right-click on Baltimore_Lights and select Edit Features > Start Editing from the context menu.
15. Ensure nothing in the map is selected, right-click on the Baltimore_Lights layer and choose Selection > Select All.
16. With the features selected, right-click anywhere in the map viewer and from the context menu select MVRsimulation Feature Definition.
17. When the MVR Polyline Feature Definition dialog box appears, click the down-arrow, and select Lights from the list. Click OK.



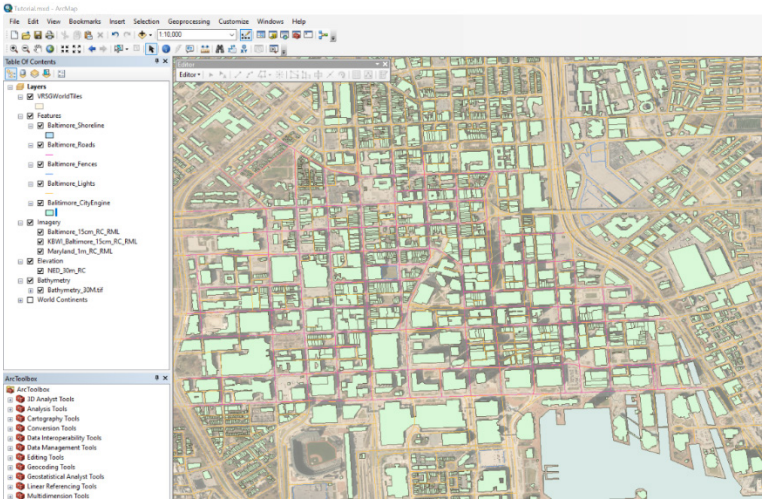
18. From the Editor menu, choose Save Edits, and then choose Stop Editing. (Due to the large number of lights, Save Edits might take a few moments to complete.)
19. Choose File > Save to save your work.

Adding buildings

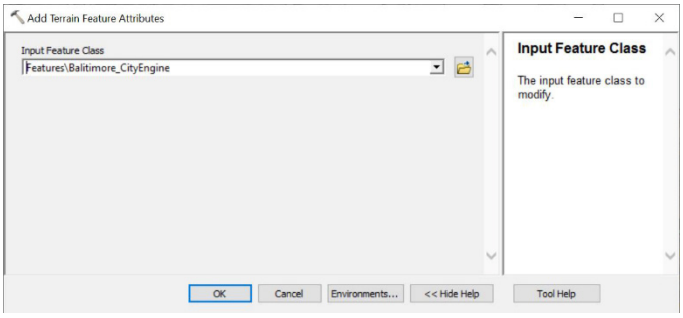
Terrain Tools contains a feature type for compiling buildings into the terrain using an Esri CityEngine rule package (*.rpk). CityEngine is a standalone Esri software product for rapidly creating procedural 3D urban models. An RPK is a compressed package, containing compiled CGA rule files (*.cgb), plus all referenced assets and data. The CGA rule files define the geometric characteristics and texturing of the procedurally made building models. The RPK used in this tutorial generates extruded buildings suitable for large coverages of buildings, such as cities. Using this feature type does not necessarily require the use of CityEngine when downloadable RPKs are available online.

Building footprints are a polygon feature type, with the geometry representing the 2D footprints of the buildings.

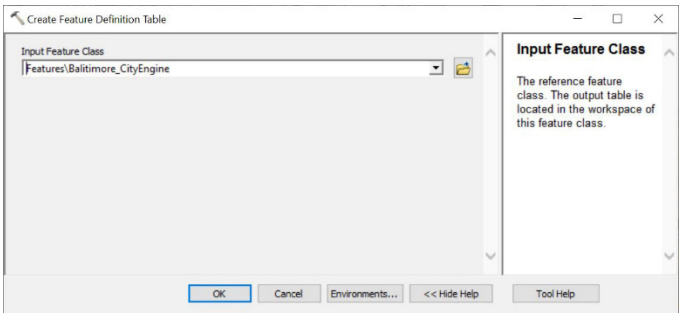
1. In the Catalog window, select and expand the Vector Data folder from the Sample Project folder. (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data)
2. Within the Vector Data folder, select and expand the \CityEngine folder to select the Baltimore_CityEngine.shp shapefile of building footprints.
3. Drag Baltimore_CityEngine.shp from the Catalog to the TOC and place it under the Features layer.
4. In the TOC, right-click on the Baltimore_CityEngine layer and select Zoom to Layer so you can see building footprints.



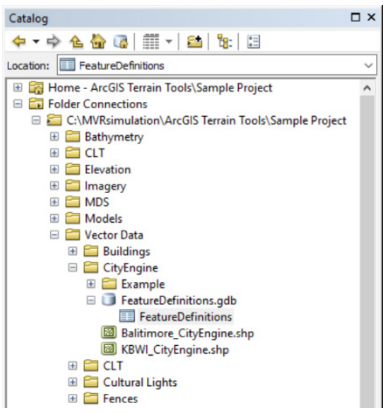
5. From the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on Add Terrain Feature Attributes.
6. When the Add Terrain Feature Attributes dialog box appears, click the down-arrow in the Input Feature Classes field, select Baltimore_CityEngine, and then click OK.



7. In the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on the tool Create Feature Definition Table.
8. When the Create Feature Definition Table dialog box appears, in on the down-arrow in the Input Feature Class field, select Baltimore_CityEngine, and then click OK.

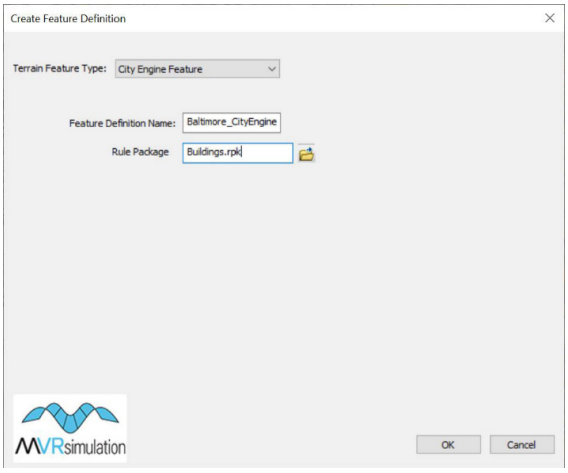


9. In the Catalog window, click on the CityEngine folder in your sample project which is located in MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\CityEngine. Right-click on this directory and select Refresh from the context menu. You should now see a FeatureDefinitions.gdb.
10. Click on the + icon to expand the contents of the FeatureDefinitions.gdb to view the feature definitions table.



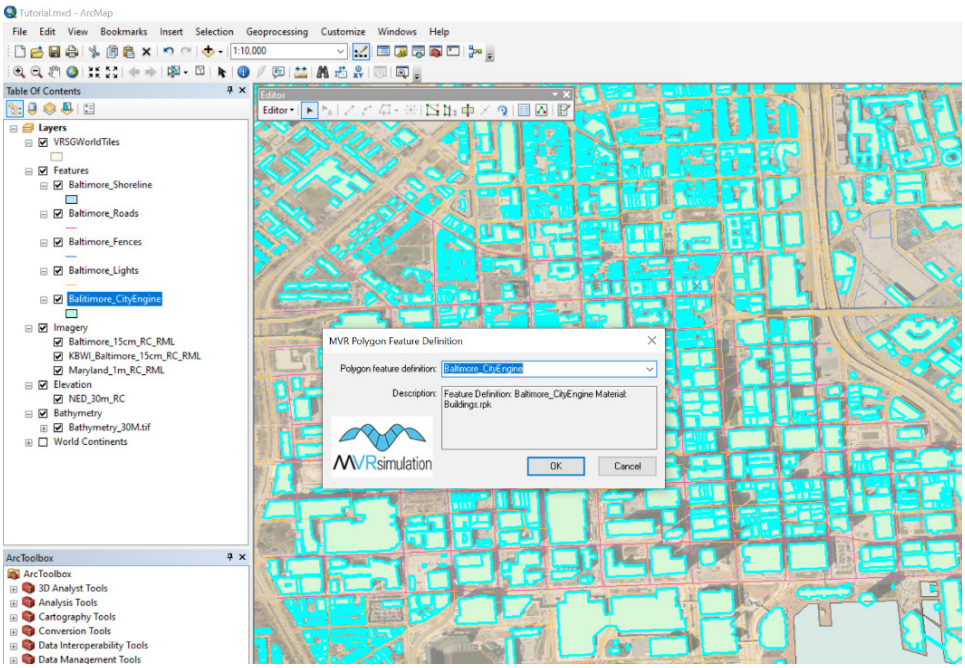
11. Right click on the Feature Definitions table, select Create MVR Feature Definition from the context menu.
12. When the Create Feature Definition dialog box appears, fill in the following information:

    - For Feature Definition Name, type “Baltimore_CityEngine”.
    - For Rule Package, browse \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\CityEngine, locate and select the file Buildings.rpk.



13. Click OK.
14. In the TOC, right-click on Baltimore_CityEngine and select Edit Features > Start Editing from the context menu.

15. Ensure nothing in the map is selected, right-click on the Baltimore_CityEngine layer and choose Selection > Select All.
16. With the features selected, right-click anywhere in the map viewer and from the context menu select MVR Feature Definition.



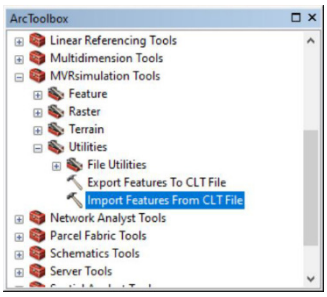
17. When the MVR Polygon Feature Definition dialog box appears, click the down-arrow, select Baltimore_CityEngine from the list, and then click OK.
18. From the Editor menu, choose Save Edits, and then choose Stop Editing.
19. Choose File > Save to save your work.

As an alternative to the CityEngine RPK workflow described above, you can also choose to export your own building models from CityEngine as described in the appendix "Adding CityEngine Generated Models to 3D Terrain."

Compiling models from a cultural feature file (.clt) into the terrain

Geospecific models can be compiled into the terrain to complement the procedural models generated by CityEngine in the previous step. In this case, the tutorial has a cultural feature file (vrsg.clt) that references a geolocated model of Baltimore's Camden Yards baseball stadium. To compile this model into a terrain tile, you will need to import the vrsg.clt file:

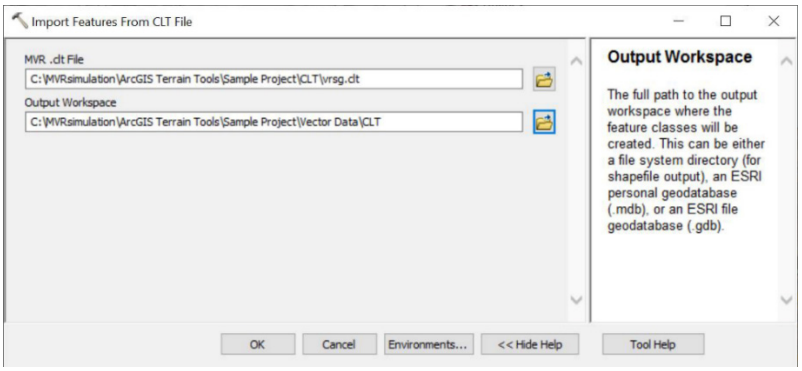
1. In the MVRsimulation Tools Toolbox, expand the Utilities toolbox and double-click Import Features From CLT File.



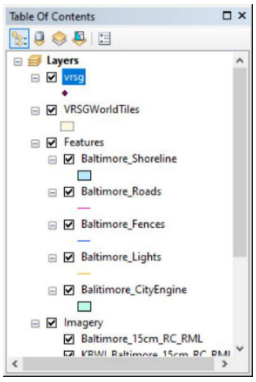
2. In the Import Features From CLT File dialog box:

   For MVR CLT File, click the browse button and locate the file vrsg.clt in \MVRsimulation\ArcGIS Terrain Tools\Sample Project\CLT.

   For Output Workspace, click the browse button and locate the folder \MVRsimulation \ArcGIS Terrain Tools\Sample Project\Vector Data\CLT.



3. Click OK to close the dialog box. After the tool finishes executing, a point feature layer called "vrsg" will appear at the top of the TOC. The point location in the map viewer is the origin of the model. *Optionally*, right-click on the layer and select zoom to layer to see its location.



4. In the TOC, right-click on the VRSG layer, and select Open Attribute Table.
5. In the TOC, right-click on the "vrsg" layer again and select Edit Features > Start Editing from the context menu.
6. Right-click on the layer and select Open Attribute Table. In the Attribute Table, notice that the Model attribute column is already prepopulated with the model's name "camden_yards.hpy". A preview of this model is shown below in Model Viewer:

Because this model is not located in the MVRsimulation 3D model libraries where Terrain Tools searches for a model by default, you will need to assign a path to the model attribute table. This workflow can be used for your own terrain projects when your project's models are located in their own separate project directory like this one. The specified model location can be an absolute path, or a relative path. For this example, you will use a relative path.

A relative path refers to a location that is relative to a current directory. Relative paths make use of two special symbols, a dot (.) and a double-dot (..), which translate into the current directory and the parent directory. Double dots are used for moving up in the hierarchy. A single dot represents the current directory itself. Consider the file locations below:

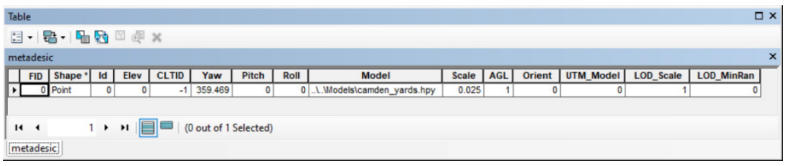
C:\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\CLT\vrsg.shp

C:\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Models\ camden_yards.hpy

To determine the relative path of the vrsg.shp file to the HPY model, you would need to go up two directories (..\..\) in order to get to the current directory. The current directory is sometimes referred to as the root directory. In this example, the Sample Project directory is the current directory. Therefore, your relative path would be ..\..\Models\camden_yards.hpy.

Note: A relative path cannot span disk drives. For example, if your current directory is on drive D: you cannot use relative paths to navigate to any directory on the E: drive. In this case, you would need to use an absolute path.

1. In the Model attribute column of the Attribute Table, assign the following relative path and model name to the Model attribute field: ..\..\Models\camden_yards.hpy

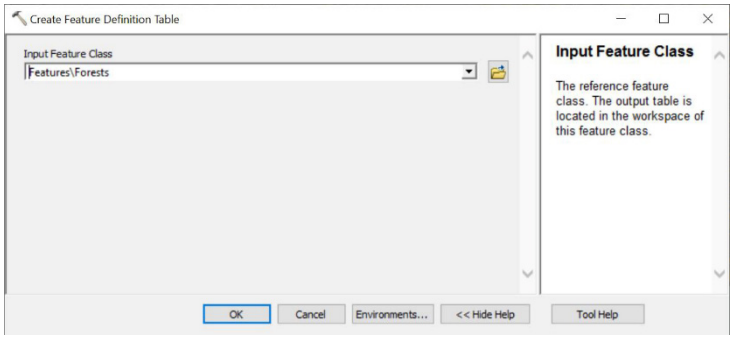


2. Close the Attribute Table and from the Editor menu, choose Save Edits, and then choose Stop Editing.
3. In the TOC, move the “vrsg” layer so that it is listed under Features. This model is now ready to compile directly into the terrain tile during the terrain build.
4. Choose File > Save to save your work.

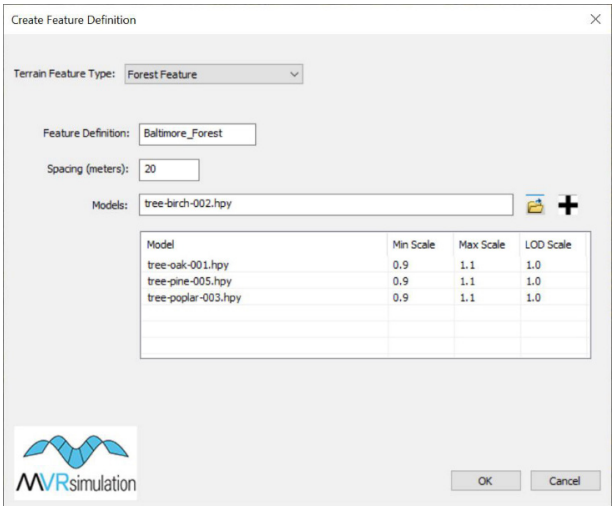
Creating areal forests

Terrain Tools v2.0 introduces a terrain feature for creating areal forests (polygon feature classes). This new feature has some advantages over the older workflow of generating points in a polygon prior to terrain compilation using the Create Forest Tool. Using this new method, Terrain Tools can partition the polygonal area into high and low detail regions. The outer region will generate high detail trees, while an inner region will produce lower detail trees for improved rendering performance for large and dense forests. In addition, polygons are much easier to work with and display in ArcMap than hundreds or thousands of points. The scale and orientations of the tree models are randomized similar to the Create Forest tool, but this happens when the terrain is compiled. In this section, you specify the polygon shapefile for your forest area, assign tree models and scale factor, and specify the density of the forest instances.

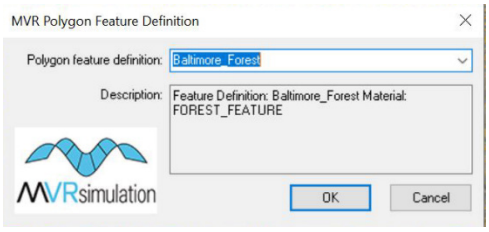
1. In the Catalog window, locate the Forests.shp from the Vector Data/Forests directory and place it in the Features layer in the TOC.
2. Next, in the Feature toolset, double-click the Add Terrain Feature Attributes tool, and in the dialog box, select the Forests polygon feature layer.
3. In the Features toolset, double-click the Create Feature Definition Table tool and again select your Forests feature layer. This will create an empty feature definition table which you will use for populating forest attributes.



4. In the Catalog window, click on the Forests directory (Vector Data/Forests). Right-click on this directory and select Refresh from the context menu. You should now see a FeatureDefinitions.gdb.
5. Click on the + icon to expand the contents of the FeatureDefinitions.gdb to view the feature definitions table.
6. Right-click on the Feature Definitions table, select Create MVRsimulation Feature Definition from the context menu.
7. When the Create Feature Definition dialog box appears, select the Forest feature type, and then enter required information. Be sure to give the Feature Definition a descriptive name so that you can easily identify it when assigning it to polygons. Click OK.



8. In the TOC, right-click the feature layer in the TOC and from the context menu choose Edit Features > Start Editing.
9. Ensure nothing in the map is selected, right-click on the Forests feature layer, and then click Selection > Select All.
10. With the features selected, right-click anywhere in the map viewer and from the context menu select MVRsimulation Feature Definition.
11. In the MVR Polygon Feature Definition dialog box, select the forest feature definition name, and then click OK.

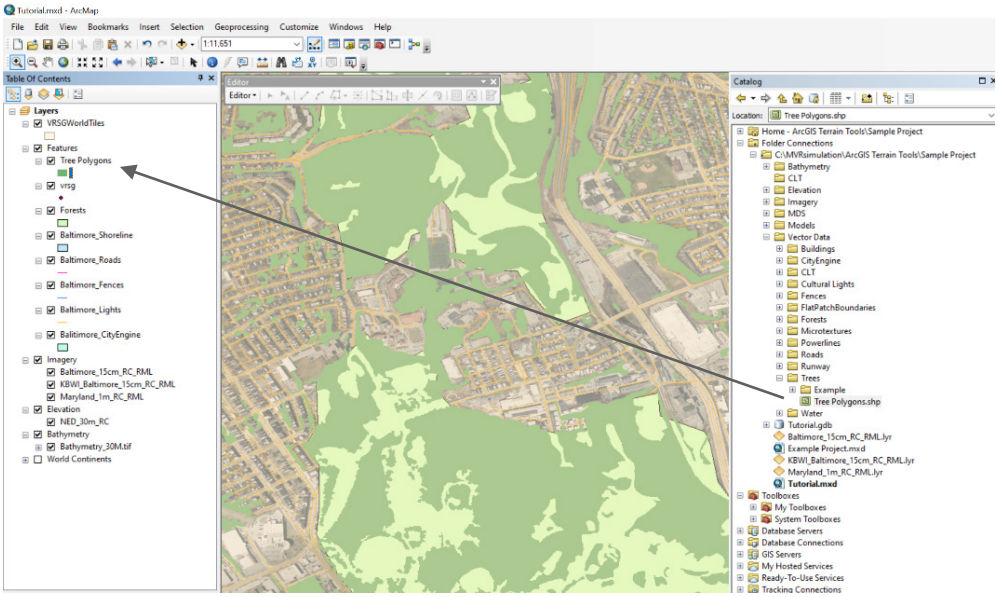


12. From the Editor menu (on the ArcMap toolbar), choose Save Edits, and then choose Stop Editing.

Creating forests with tree points

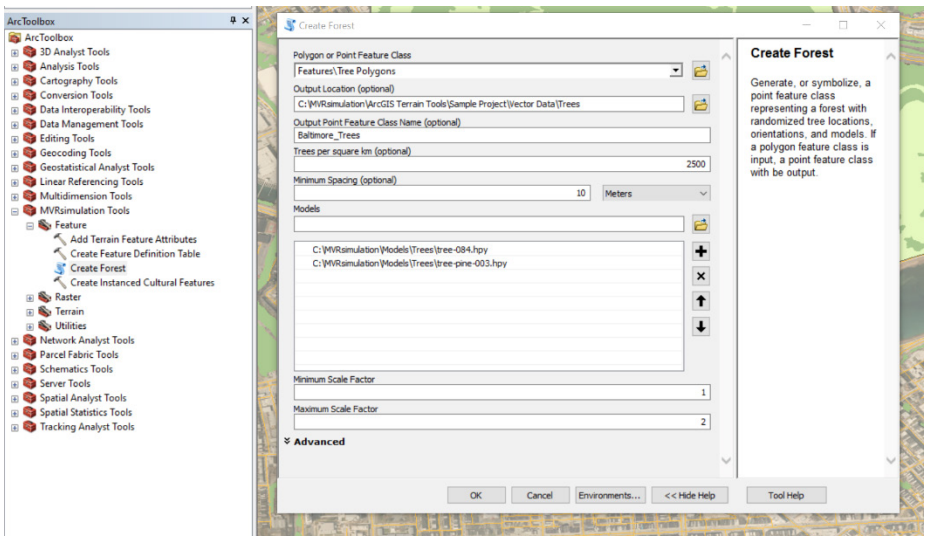
The Create Forest tool provides a quick and efficient way to distribute thousands of randomized and scaled tree points within a polygon. The Create Forest tool is also useful when assigning tree models to an existing geospecific tree point shapefile. In this section, you assign random points within a polygon feature class, assign tree models and scale factor, and specify the distribution density and minimum spacing between the trees.

1. In the Catalog window, within the Vector Data folder, select and expand the \Trees folder and locate the shapefile Tree Polygons.shp.
2. Drag Tree Polygons.shp from the Catalog to the TOC and place it under the Features layer. (Disregard the ArcMap geocoordinate system error message.)



3. In the TOC, right-click on the Tree Polygons layer and select Zoom to Layer.
4. From the MVRsimulation Tools toolbox, expand the Feature toolset and double-click the Create Forest tool.
5. When the Create Forest dialog box appears, enter the following information:
   - For Polygon or Point Feature Class, click the folder browse button and navigate to the shapefile Tree Polygons.shp located in \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\Trees.
   - For Output Location, click the folder browse button and navigate to the directory \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data and select \Trees.
   - For Output Point Feature Class Name, type “Baltimore_Trees”. This action provides the name of a new forest feature class.
   - For Trees per square km, type in the value “2500”. This value specifies the density of the forest you are creating, based on the area of each input feature. This density is limited by the Minimum Spacing parameter.

- For Minimum Spacing, change the default value of 5 meters to "10" meters. This value is the minimum allowed spacing between each tree model. This value takes precedence over the density parameter. If the points populated from the Trees per square km value would be closer than the Minimum Spacing value, the density is reduced until there are no points closer than the spacing.
- For Models, use one or more tree models from the models that are installed with VRSG. In Windows Explorer, navigate to \MVRsimulation\VRSG\Models\Trees, locate Tree-084.hpy and drag the model file to the Models field in the Create Forest tool.



- Repeat the Models step above to add the model Tree-pine-003.hpy to the list.
- For the Maximum Scale Factor, change the default value of 1 (the original scale of the tree models) to "2" to increase the random scaling of each tree model. (Keep the Minimum Scale Factor value at 1.)

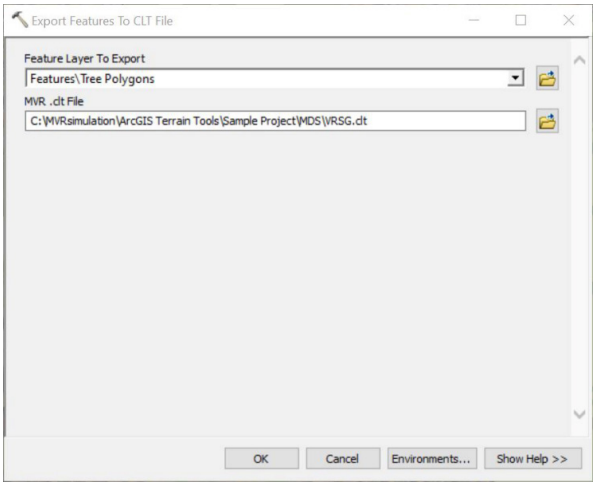
6. Click OK to close the dialog box.
7. After the Create Forest tool finishes executing, a Baltimore_Trees shapefile will appear at the top of the TOC, and the point features should appear in the map viewer.
8. In the TOC, click on the Baltimore_Trees layer and drag it to a position under the Features layer.
9. Choose File > Save to save your work.
10. Now that you have generated a tree point layer from your tree polygon layer, you can unselect or remove the Tree_Polygons layer from the TOC. To remove the layer, right-click the Tree_Polygons layer and select Remove.

Incorporating these tree models in the terrain database can be achieved in a couple of ways. You can compile the tree models directly into the terrain tiles by simply placing Baltimore_Trees layer under the Features layer in the TOC. The tree model locations are represented by tree points in the map viewer will compile the assigned tree models directly into the terrain tiles.

Alternatively, you can export the point feature data to a cultural feature file (vrsg.clt file), which VRSG will load at runtime when it is added to the VRSG search path directory.

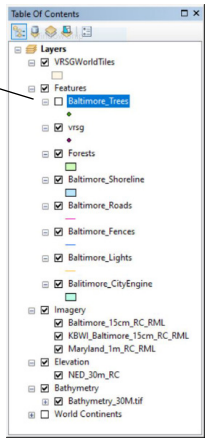
To export the point feature data to a cultural feature file (vrsg.clt):

1. In the MVRsimulation Tools toolbox, expand the Utilities toolbox and double-click the tool Export Features to CLT File.
2. When the Export Features to CLT File tool appears, for Feature Layer to Export, click the down-arrow and then select Baltimore Trees from the list.
3. For MVR CLT File, click the browse button and navigate to: \MVRsimulation\ArcGIS Terrain Tools\Sample Project\MDS\. In the browse dialog box, for File Name, type "vrsg.clt" and then click Save.



4. Click OK to close the Export Features tool.
5. In the TOC, unselect the checkbox next to the Baltimore_Trees layer to disable that layer. As the trees now reside in a vrsg.clt cultural feature file for loading at VRSG runtime, you do not want them to be compiled in the terrain.

Unselect this checkbox.



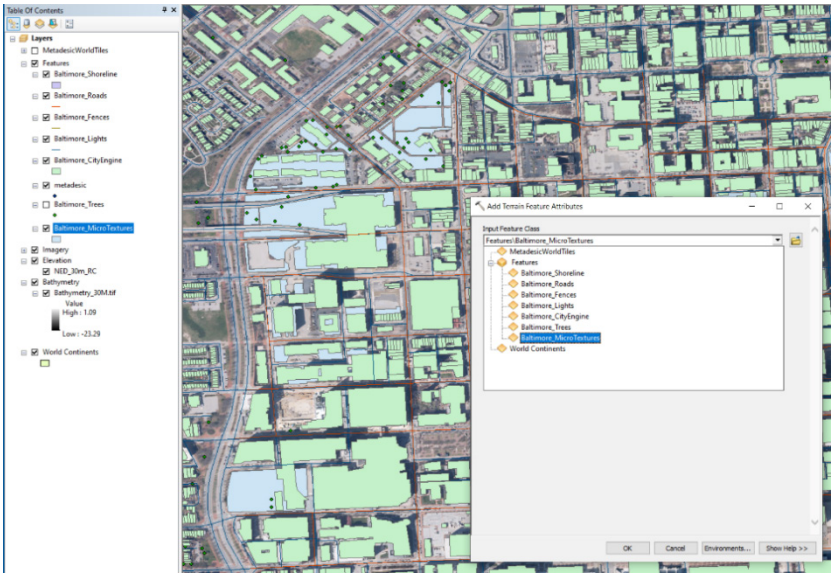
6. Choose File > Save to save your work.

Adding microtextures

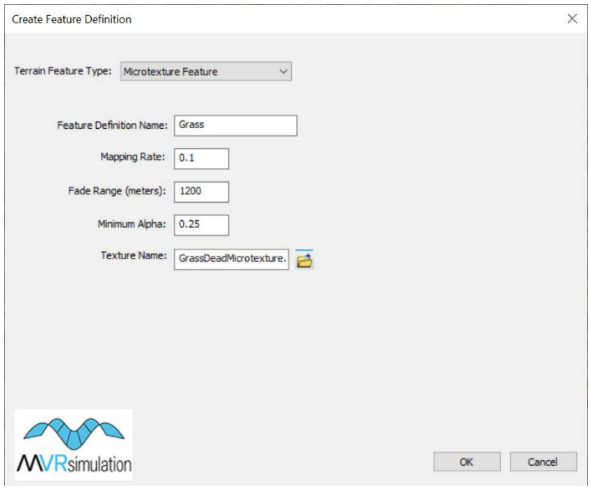
Microtextures are geotypical ground textures, such as soil, grass, or sand textures, which are added to the terrain to enhance the realism of the ground surface when it is viewed at close range in a ground-based simulation. To use one overall microtexture with a terrain you can simply supply one for VRSG to load a runtime (as described in the chapter "Visualizing 3D Terrain in VRSG." But in Terrain Tools, you can have multiple microtextures compiled in the terrain for specific areas, such a grass, sand, or snow. In these next steps, you will add a single grass microtexture to a polygon layer. The chapter "Working with Terrain Features" describes how to add multiple microtextures to the terrain.

In the Catalog window, locate the Sample Project folder and then select and expand the Vector Data folder (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data).

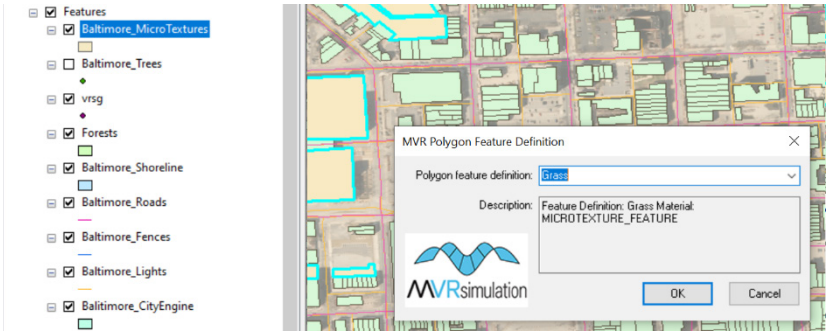
1. Within the Vector Data folder, select and expand the \Microtextures folder and locate the shapefile Baltimore_MicroTextures.shp.
2. Drag Baltimore_MicroTextures.shp from the Catalog to the TOC and place it under the Features layer.
3. Right-click on the Baltimore_MicroTextures layer and choose Zoom to Layer to see where the microtexture will be applied.
4. From the MVRsimulation Tools toolbox, expand the Feature toolset and double-click the Add Terrain Feature Attributes tool.
5. In the Add Terrain Feature Attributes dialog box, for the Input Feature Class field, click the down arrow and select Baltimore_MicroTextures, and then click OK.



6. From the MVRsimulation Tools toolbox, expand the Feature toolset and double-click the Create Feature Definition Table tool.
7. In the Create Feature Definition Table dialog box, for the Input Feature Class field, click on the down arrow, select Baltimore_MicroTextures from the list, and then click OK.
8. In the Catalog window, click on the Microtextures folder in your sample project. Right-click on this directory and select Refresh from the context menu. You should now see a FeatureDefinitions.gdb.
9. Click on the + icon to expand the contents of the FeatureDefinitions.gdb to view the feature definitions table.
10. Right click on the Feature Definitions table, select Create MVRsimulation Feature Definition from the context menu.
11. In the Create Feature Definition dialog box, fill in the following information:
    - For Terrain Feature Type, select Microtexture Feature.
    - For Feature Definition Name, type “Grass”.
    - For Texture Name browse to \MVRsimulation\ArcGIS Terrain Tools\Textures \Ground and select GrassDeadMicrotexture.tex, and then click OK. (Leave the other values in the dialog box as they are.)



12. In the TOC, right-click on Baltimore_MicroTextures and select Edit Features > Start Editing from the context menu.
13. Ensure nothing in the map is selected then right-click on Baltimore_MicroTextures layer and select Selection > Select All.
14. With the features selected, right-click anywhere in the map viewer and from the context menu select MVRsimulation Feature Definition.
15. When the MVR Polygon Feature Definition dialog box appears, select Grass, and then click OK.



16. From the Editor menu, choose Save Edits, and then choose Stop Editing.
17. Choose File > Save to save your work.

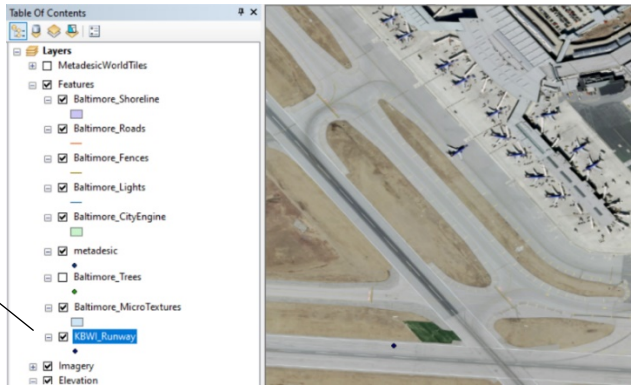
Adding an airport runway model

Textured airport runway models are often generated in an external modeling tool as a flat (2D) mesh. These models can be compiled directly into the 3D terrain as a runway feature. Runway models are incorporated as point features representing the origin and placement of the model. When a flat runway model is compiled into the terrain, it will conform to the earth's curvature and terrain profile.

To add an existing runway model into the terrain:

1. Within the Vector Data folder, select and expand the \Runway folder, and drag the shapefile KBWI_Runway.shp from the Catalog window to the TOC. Place it under the Features layer.

Runway point feature in ArcMap.

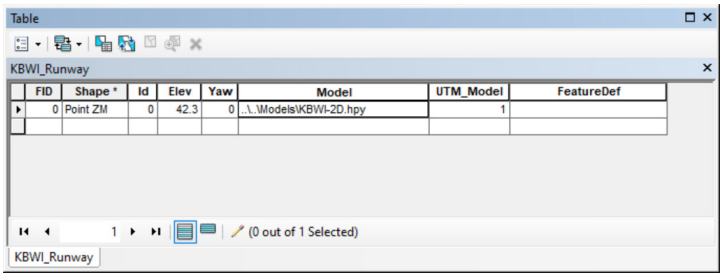


2. From the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on Add Terrain Feature Attributes.
3. When the Add Terrain Feature Attributes dialog box appears, click the down-arrow in the Input Feature Classes field, select KBWI_Runway, and then click OK.
4. In the TOC, right-click on the KBWI_Runway layer and choose Zoom to Layer to see the location of the airport runway model.
5. Right-click on KBWI_Runway, and select Edit Features >Start Editing.

6. Right-click on the layer and select Open Attribute Table. In the Attribute Table, notice that the Model attribute column is already prepopulated with the model's name "KBWI-2D.hpy".

   You will need to edit that field and add a relative path to this model field just as you did for the Camden Yards model in the previous steps since the model is located outside of the VRSG Model library.

7. In the Model attribute field of the Attribute Table, assign the following relative path to the runway model in the attribute field: ..\..\Models\KBWI-2D.hpy

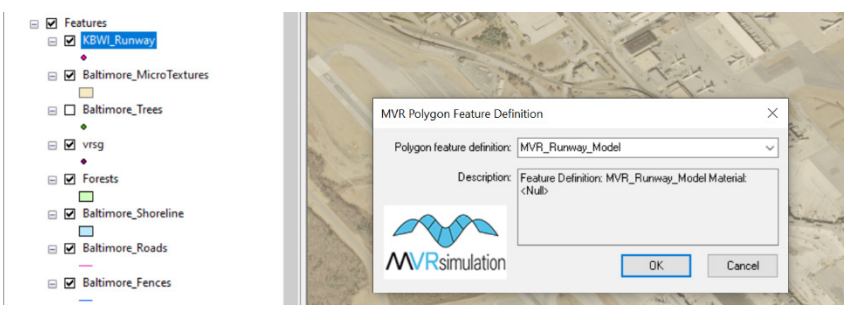


8. Close the Attribute Table and from the Editor menu, choose Save Edits.

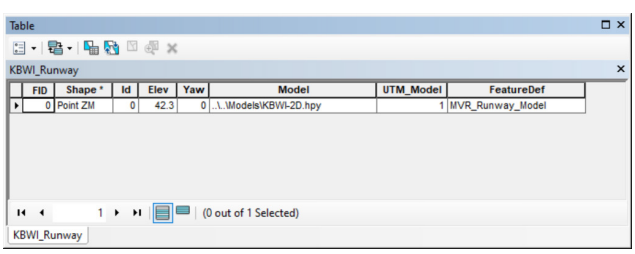
9. Ensure nothing in the map is selected, right-click on the runway layer, and click Selection
   > Select All.

10. With the features selected, right-click anywhere in the map viewer and from the context menu select MVRsimulation Feature Definition.

11. When the MVR Polygon Feature Definition dialog box appears, click the down-arrow, select MVR_Runway_Model from the list, and then click OK.



12. Open the attribute table and verify the FeatureDef is populated with the MVR_Runway_Model feature definition. Close the attribute table.

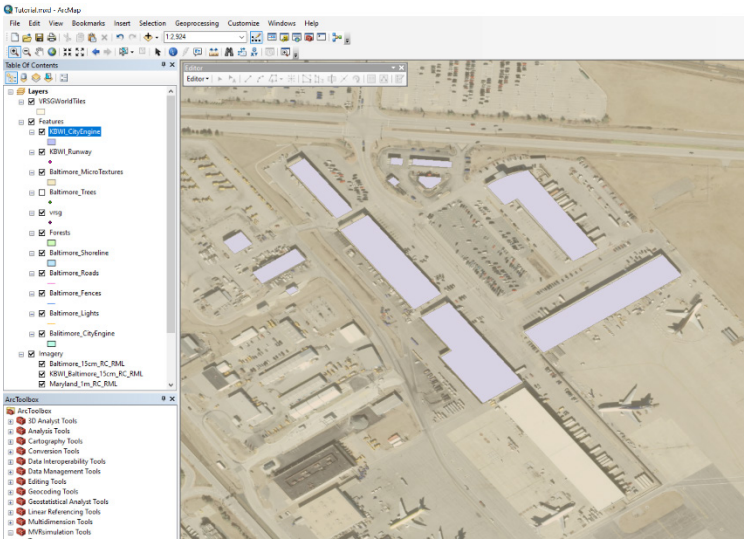


13. From the Editor menu, choose Save Edits, and then choose Stop Editing.
14. Choose File > Save to save your work.

Adding airport buildings

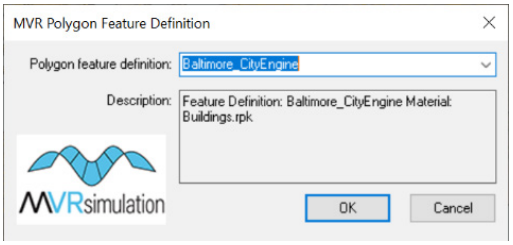
Lastly, before building the tiles, you can add a set of buildings by the airport. Like the building models of downtown Baltimore, they are included in the CityEngine rule package (Buildings.rpk).

1. In the Catalog window, select and expand the Vector Data folder from the Sample Project folder (\MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data).
2. Within the Vector Data folder, select and expand the \CityEngine folder to select the KBWI_CityEngine.shp shapefile of building footprints.
3. Drag KBWI_CityEngine.shp from the Catalog to the TOC and place it under the Features layer.
4. In the TOC, right-click on the KBWI_CityEngine layer and select Zoom to Layer so you can see building footprints.



5. From the MVRsimulation Tools toolbox, expand the Feature toolset and double-click on Add Terrain Feature Attributes.
6. When the Add Terrain Feature Attributes dialog box appears, click the down-arrow in the Input Feature Classes field, select KBWI_CityEngine, and then click OK.
   Unlike earlier, you do not need to create a feature definition table for these airport CityEngine buildings, as you already created one for the downtown Baltimore CityEngine buildings. This KBWI shapefile will use the same feature definition table and Buildings.rpk.
7. Right-click on KBWI_CityEngine, and select Edit Features > Start Editing.

8. Ensure nothing in the map is selected, right-click on the KBWI_CityEngine layer, and click Selection > Select All.
9. With the features selected, right-click anywhere on the map viewer and from the context menu select MVRsimulation Feature Definition.
10. When the MVR Polygon Feature Definition dialog box appears, click the down-arrow, select Baltimore_CityEngine from the list, and then click OK.

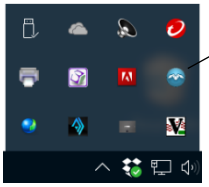


11. From the Editor menu, choose Save Edits, and then choose Stop Editing.
12. Choose File > Save to save your work.

Building VRSG terrain tiles

Before building the terrain tiles:

- Save the map document.
- Make sure the Build Manager is running on your machine by locating the MVRsimulation application icon in the Windows system tray.



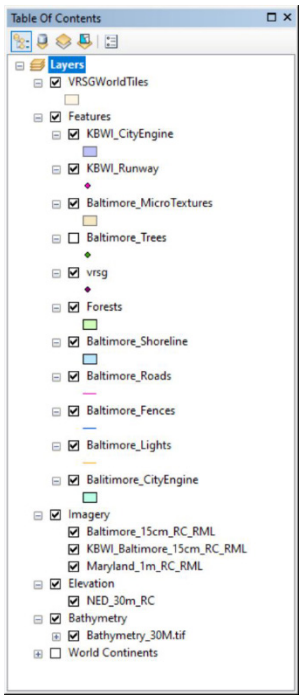
Icon for the Terrain Tools Build Manager.

If you have rebooted your machine since installing Terrain Tools, the Build Manager process will have automatically started, and should be running in the background. If it is not running, start it from Start Menu > All Programs > MVRsimulation > Terrain Tools Build Manager.

Once the Build Manager is enabled, the BuildManager.exe should be visible in your system's Task Manager under Background Processes.

The Build Manager is launched by the Build Selected MDS Tiles tool in the Terrain toolset, as shown below. This tool examines all features currently selected in the map and determines whether any of them represent VRSG terrain tiles. If so, this tool instructs Build Manager to build terrain tiles from selected tiles on the map using the map contents as source data.

Before you compile the terrain tiles for the Sample Project, review the example TOC below.



Ensure that your layers are organized under the appropriate group categories. The order of the features listed under the Features group layer is not important, however, ensure that the imagery layers are organized in the same order as the example below with the highest resolution imagery placed at the top of the stacking order.

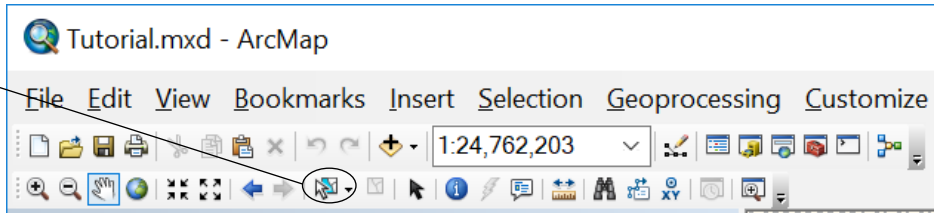
Building selected tiles from the map

You select the set of tiles you want to build from the VRSGWorldTiles layer. If you had hidden the display of this layer previously, make it visible now, by selecting its checkbox in the TOC.

To build a set of terrain tiles:

1. In the TOC, right-click on Maryland_1m_RC_RML and choose Zoom to Layer. In this view, you can see both downtown Baltimore and the BWI airport imagery. Notice the triangles from the VRSGWorldTiles layer demarking where the boundaries of each terrain tile.
2. Right-click on the VRSGWorldTiles layer and choose Selection > Make this the Only Selectable Layer. *This step is critical.* (By default, all feature layers are selectable. This means that if you select tiles from the VRSGWorldTiles layer without making it the only selectable layer, you could potentially also select features from other layers.)
3. On the ArcMap toolbar, click the Select Features tool (button with the white arrow and light blue triangle).

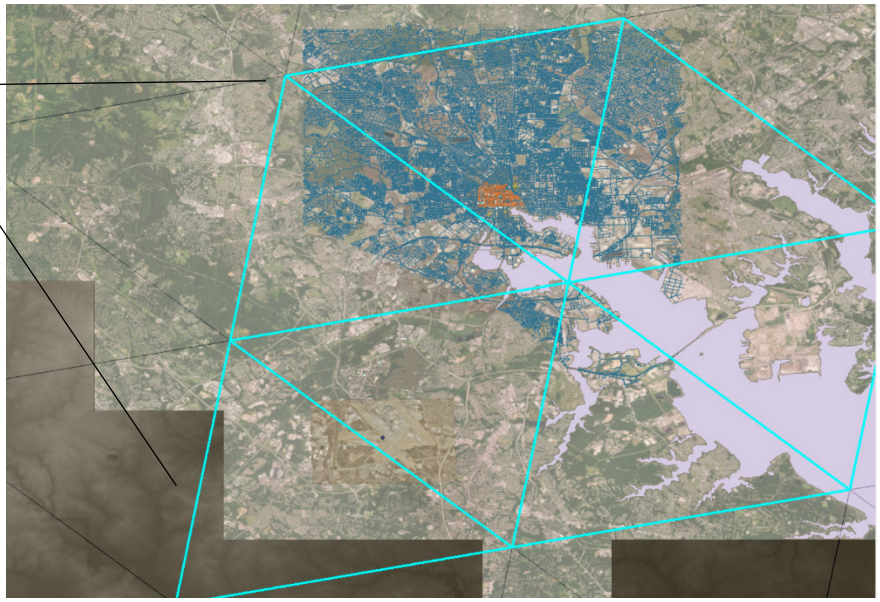
ArcMap Select Features Tool.



This tool enables you to click to select a feature or drag a selection box to select multiple features (in this case, tile boundaries).

4. Select six or seven triangular tiles in the VRSGWorldTiles layer, as shown in the example below. Make sure your selection includes at least some of both downtown Baltimore and the BWI airport. (Notice that the airport imagery falls across two tiles and that one of those tiles is not fully covered by the imagery.)

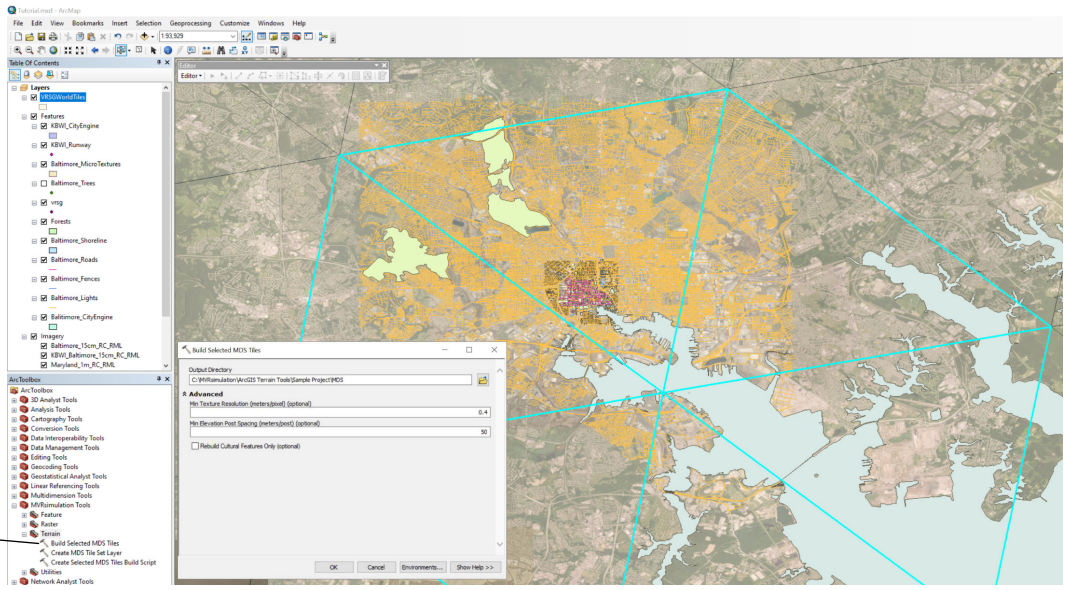
Highlighted boundaries of selected terrain tiles that will be built.



To select each tile individually, press and hold the Shift key as you click each tile. Alternatively, to select them all at once, hold down the left mouse button to draw a selection rectangle over the tiles. As you select them, the boundaries of the potential tiles will appear with a light blue outline, as shown above, to indicate that they are selected.

(If you selected some tiles you do not want, right-click and choose Clear Selected Features to unselect the tile boundaries, and then start fresh.)

5. Expand the MVRsimulation toolbox, and in the Terrain toolset, double-click the Build Selected MDS Tiles tool.



Double-click the Build Selected MDS Tiles Tool.

6. When the Build Selected MDS Tiles dialog box appears, for the Output Directory click the folder browse button and navigate to \MVRsimulation\ArcGIS Terrain Tools\Sample Project\MDS directory. This directory will store the six terrain tiles that will be built.

7. Optionally, click the Advanced down-arrow on the dialog box, and adjust the minimum texture resolution or post spacing for the tiles to be generated. (You can also just leave the default values as they are.)

   The resolution of the terrain tiles is driven by the underlying source data. The default values provided for these fields are sufficient for most situations. However, in some cases, you might want to build the terrain tiles to a resolution that is not as fine as the source data, if the full resolution of the source data is not needed in the final terrain, or as a way to speed up tile building during terrain development.

   - The minimum texture resolution setting defines a lower bound on the resolution of the terrain textures that will be generated. By setting this value (in meters), you determine that the source imagery will not be sampled at a finer resolution than the given value during the terrain build.
   - In the same way, the minimum post spacing setting defines the minimum distance between terrain vertices in the output terrain. If features are added to the terrain, they might cause vertices to be denser than the value of this setting would indicate, but the elevation source data will not be sampled at a finer resolution than the value provided.

8. Click OK to close the dialog box.

Once you have started the process of building the terrain tiles, ArcMap will prompt you to save the map.

Next, the Terrain Tools Build Manager appears, displaying the progress of the tile-building process. The Build Manager has a browser interface with navigation links to pages located on the left side of the window.

Note: If the Build Manager is not running as a background process (for example if this is the first time you have used Terrain Tools), you might need to start it manually. To do so, locate "Terrain Tools Build Manager" in the Windows Start menu and double-click it.

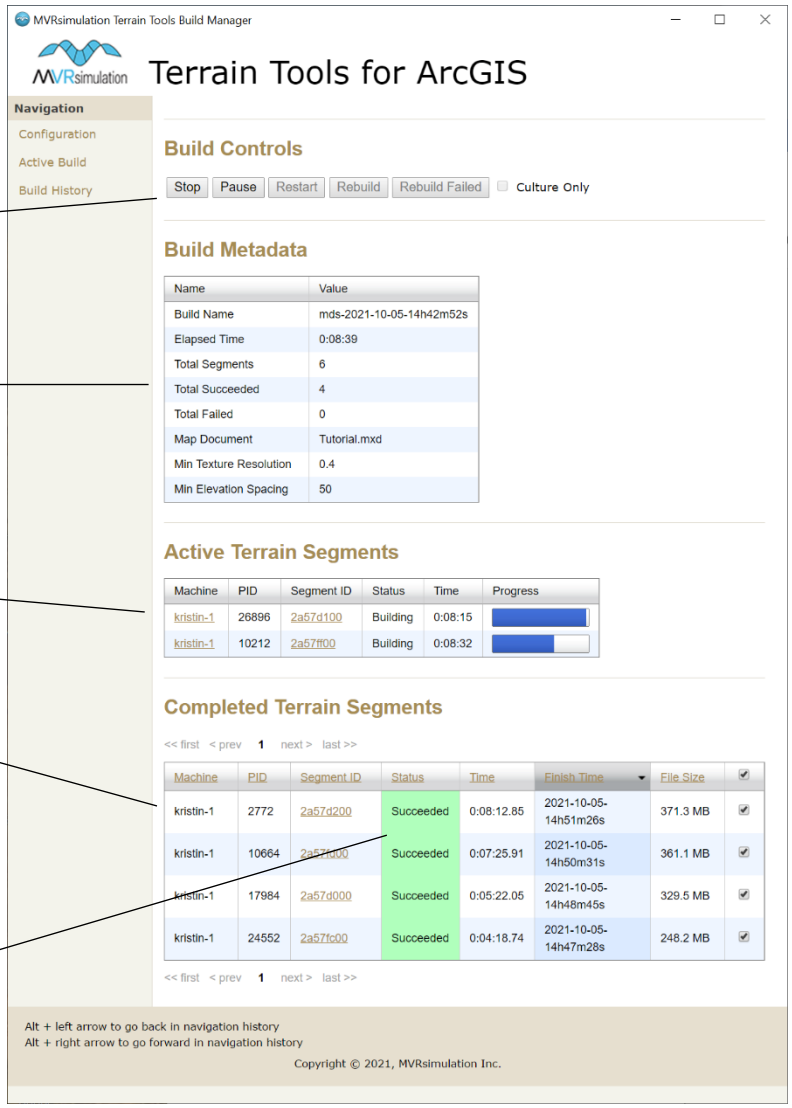
Buttons to stop, pause, resume, or rebuild the active terrain-building process.

General information about the active terrain building process.

Information about all the active terrain-building processes.

As each tile is built, information about that completed tile appears in a row in this table.

Green indicates a completed tile, yellow indicates an incomplete tile (in process or a paused process), and red indicates a failed tile process.



In the Navigation panel, click the Active Build link, which displays a page reporting some information about the terrain-building process, the status of each tile as it is being built, and the tiles that have been completed.

As each tile is built, information about that completed tile appears in a row in the Completed Terrain segments table.

While the Build Manager is running, you do not need to keep the Tutorial map document in ArcMap open; you can close ArcMap.

The Terrain Tools Build Manager window can be used for configuring and managing a build server with one or more multiple terrain-building client machines. (For this tutorial, the machine you have been using is treated as both a server and client.) These activities are described in full in the chapter, "Managing Terrain Builds."

When the terrain tiles have all been built, close the Build Manager.

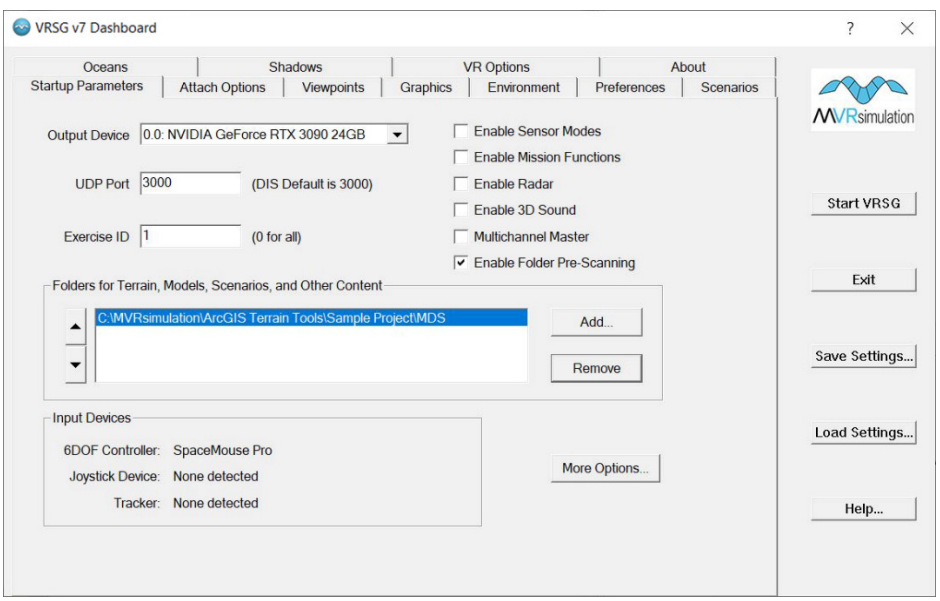
If a tile fails to build, click on the tile name ("Segment ID") to display a log with information about the processing of that tile.

Viewing terrain tiles in VRSG

Once one or more terrain tiles have been built, you can view them in VRSG.

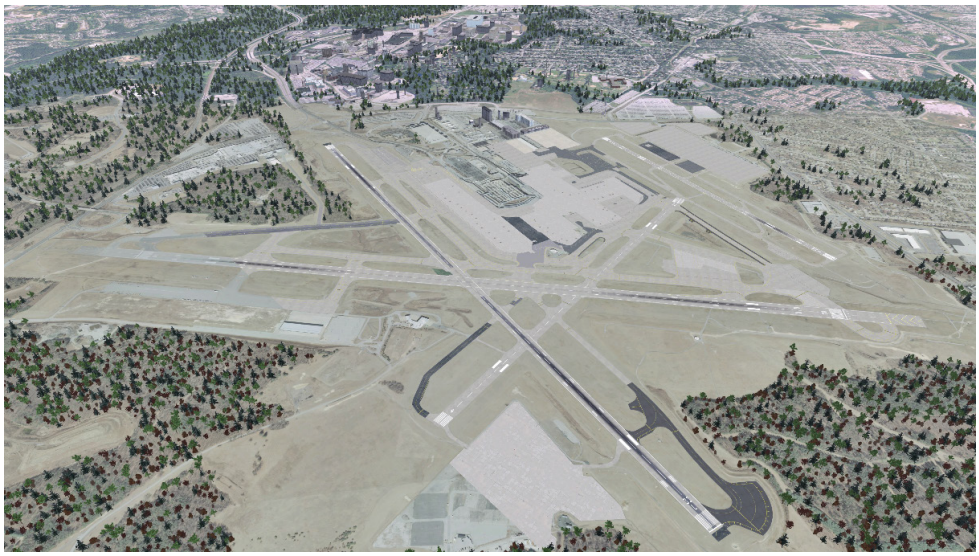
To visualize the 3D terrain in VRSG:

1. From the MVRsimulation program folder on the Start menu, choose VRSG 7.
2. When the VRSG Dashboard appears, on Startup Parameters tab, in the "Folders for Terrain, Models, Scenarios and Other Content" section, click Add and browse to the folder \MVRsimulation\ArcGIS Terrain Tools\Sample Project and then select the \MDS folder to specify the directories of the terrain tiles, models, and textures you want to visualize. This is called VRSG's *search path*.



Note: Any textures used for features or microtextures must be in VRSG's search path, in a \Textures subdirectory of a terrain tiles directory you specified. You do not need to explicitly specify the \Textures subdirectory as you did for the \MDS directory. VRSG will load all textures it finds in the subdirectories of the tile directories. However, be sure to specify the \Sample Project\Models directory which contains the BWI runway model.

3. Click Start VRSG. When the terrain is loaded in VRSG's visualization window, initially it might display water. You would have to fly around with the 6DOF controller to locate the terrain, as you have not yet set any viewpoints to direct VRSG which area to display.
4. Open the Windows File Explorer and navigate to the \MVRsimulation\ArcGIS Terrain Tools\Sample Project\MDS directory. Select a terrain tile (.mds file) and drag it to the VRSG visualization window. (For example, tile 2a57ff00.mds.) This action causes the VRSG eyepoint to move to the center of that terrain tile.
5. In VRSG, move around the terrain using the 6DOF controller until you find the areas of downtown Baltimore and the BWI airport.



See the chapter "Visualizing 3D Terrain in VRSG" for more information about using VRSG.

CHAPTER 3

Refining Terrain Construction

This chapter introduces techniques that you can use to improve the visual appearance of the output terrain. Techniques include how to use a Raster Mosaic Layer to enhance the appearance of your imagery mosaics to achieve a seamless transition between different imagery datasets. In addition, this chapter describes:

- How to mask regions with NoData or other undesirable areas in your imagery mosaics.
- How to create an MDS tile set layer to visualize the tile coverage of VRSG terrain tiles (.mds) within a given directory.

Creating a Raster Mosaic Layer

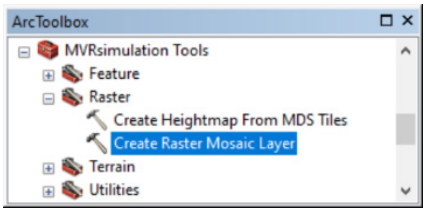
The Raster Mosaic Layer (RML) is an MVRsimulation layer that can seamlessly render raster mosaics. Working with an RML you can modify the appearance of a raster layer without changing the underlying source raster data; it merely modifies how the rasters are rendered in a map document. For example, in an RML layer you can adjust the brightness and contrast of a mosaic using the Effects toolbar, modify the RGB levels of a mosaic by applying a gamma stretch. RMLs also enable you to add a feather blend region to the mosaic, which is useful when you are attempting to create a seamless transition between different imagery layers. Because of these benefits, MVRsimulation strongly recommends that you always use Raster Mosaic Layers in your terrain projects.

The underlying data type for a Raster Mosaic Layer is a raster catalog, which is a built-in ArcGIS data type. A raster catalog is a simple container for managing raster datasets and is stored within a geodatabase. When you create a raster catalog you can then add one or more raster datasets to your empty raster catalog. MVRsimulation recommends the use of *unmanaged* raster catalogs which simply point to existing raster files on disk that ArcGIS can read. The rasters are not copied and stored in the raster catalog as they are when they are managed. This allows for faster processing times. For more general information about the raster catalog type, see the Esri ArcGIS Desktop Help.

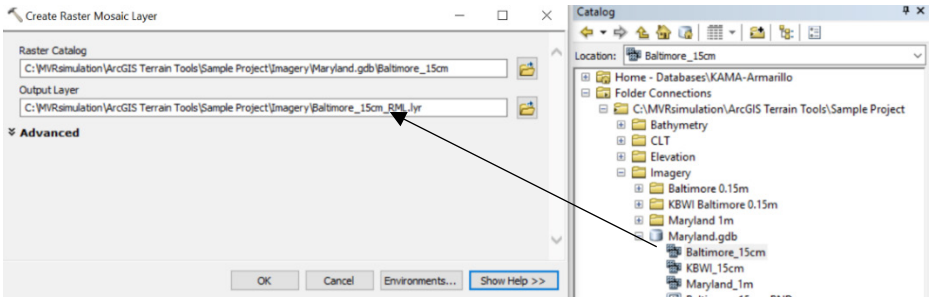
How to create a Raster Mosaic Layer

To create a Raster Mosaic Layer (RML) in ArcMap:

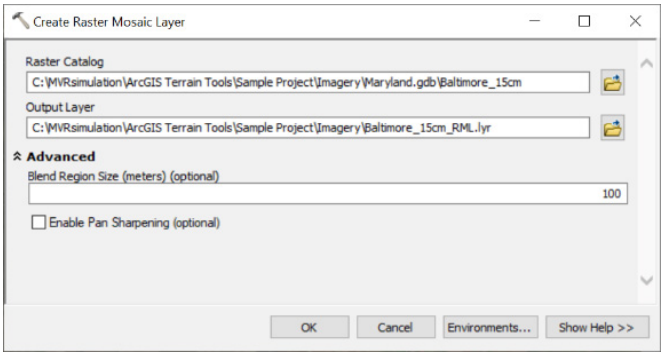
1. In the Raster toolset (located in the MVRsimulation Tools toolbox), double-click the Create Raster Mosaic Layer tool.



2. In the Raster Catalog parameter, add a raster catalog containing the raster data for the mosaic. You can easily drag a raster catalog from the Catalog window to the Raster Catalog parameter. This will automatically set the Output Layer parameter based on the location of the raster catalog. (Rename and/or change the layer to a new output path if necessary).



3. Optionally, expand the Advanced parameter to specify a blend region in meters. (The blend region can also be added and modified after the creation of the RML. To do so, right-click on the layer and select properties. Under the Raster Mosiac tab, edit the blend region parameters.)



4. Optionally, select the Enable Pan Sharpening parameter. This option is typically used for panchromatic (B&W) imagery paired with underlying RGB imagery. (You can also assign this parameter after the creation of the RML. Pan sharpening is described later in this chapter.)

   When the layer is completed, the RML will automatically appear at the top of the TOC. (Additionally, a .lyr file will appear on disk at the specified output location. Move the RML by dragging the layer under the Imagery group layer in the TOC.

   The order of data listed under the Imagery group (and Elevation group) is significant as the order affects terrain compilation. Terrain Tools compiles imagery and elevation data according to the order they are listed in the TOC layers. The order of the feature data in the TOC is not important; it does not affect the terrain compilation.

5. Once you have created Raster Mosaic Layers for all of your Raster Catalogs, remove any raster catalogs from the TOC, as you no longer need them.

Using a Raster Mosaic Layer to make visual adjustments

Using a Raster Mosaic Layer, you can modify the appearance of a layer to make it more visually appealing. Several different rendering techniques can be used to achieve a more seamless appearance between imagery datasets. In this section, you will learn how to use a Raster Mosaic Layer to make these visual adjustments.

Masking regions with no data

A common difficulty with working with geospatial raster data is handling missing data values which are often found in many spatial datasets. A NoData region is a region of an image that is usually marked by an absence of data.

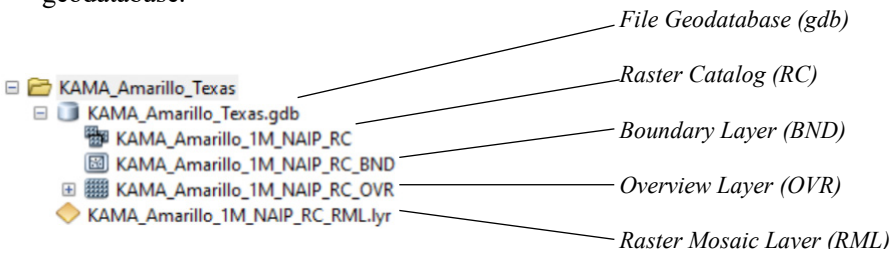
The Raster Mosaic Layer provides a simple approach to masking these NoData values. Since a raster catalog stores a footprint polygon for each raster in the catalog, there is a built-in association of polygons to rasters that can be leveraged for masking. By default, the footprint polygons that are stored by a raster catalog are initialized to be the full rectangular bounds of each raster. But once the Raster Mosaic Layer has been created, a polygonal boundary file (BND) representing the extents of a mosaic can be modified. This means that you can edit this polygon to mask out any NoData pixels.

Since the BND layer is the full rectangular extent of its associated raster dataset, the full raster dataset will be rendered in a mosaic if the BND layer is not modified. But if the BND layer has been modified to cover only a portion of the raster dataset, the portions of the raster dataset that fall outside of the BND layer will not be drawn. Modifying the BND layer to achieve NoData region masking can be done in one of two ways:

- Manually – You can manually edit this BND layer in ArcMap during an Edit session.
- From existing data – Sometimes metadata included with source imagery will contain shapefile representations of the desired raster mask polygons. In this case, you would want to use this data as-is instead of manually modifying raster footprints.

To modify the BND layer manually:

1. After creating a Raster Mosaic Layer, in the Catalog window, navigate to its associated raster catalog which is located inside a file geodatabase, and expand the contents of the geodatabase.



2. With the Raster Mosaic Layer in the map document, select the BND layer from the Catalog window and add it to the TOC above the RML.
3. Right-click on the BND layer, select Edit Features, Start Editing.
4. Right-click on the BND layer, select Properties, select the symbology tab, and change the symbol to Hollow, and click OK. You should now be able to see your imagery with the BND layer positioned above.
5. Right-click on the BND layer, select Make this the only Selectable Layer.
6. Select the BND layer in the map viewer.
7. With the BND layer selected, select the Reshape Feature Tool and begin editing the BND layer by excluding unwanted pixel areas.
8. Once complete, from the Editor toolbar, choose Save Edits and Stop Editing.
9. Remove the BND and RML layers from the Map document.
10. In the Catalog window, delete the RML layer. (You can also delete this layer in Windows Explorer.)
11. In the Catalog window, delete the associated Overview (OVR) layer. (Do not delete the BND.)
12. In the MVRsimulation Toolbox, expand the Raster Toolbox and select the Create Raster Mosiac Layer tool.
13. Re-add the same Raster Catalog that you initially added when creating your initial RML and click OK.
14. The new RML layer will appear in the TOC. Drag the RML under the Imagery group layer.
15. Inspect the RML layer in the map viewer to ensure the unwanted areas are excluded.
16. Save the map document.

To edit using existing mask polygons:

1. Copy the polygon shapefiles into the same directory as the images.
2. Name the shapefiles with the same name as their associated imagery file (such as ImageFile1.tif and ImageFile1.shp).

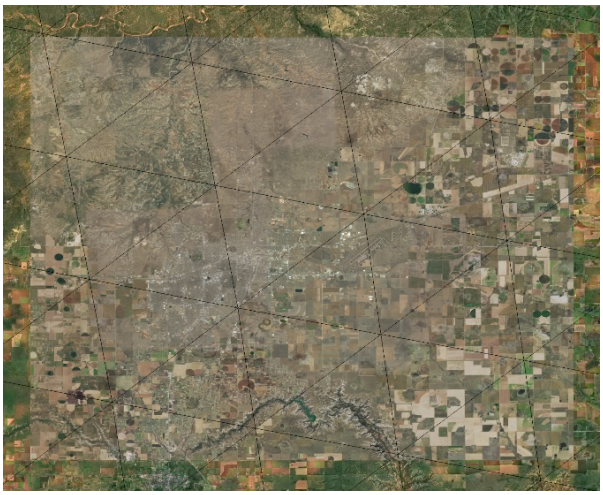
When the Create Raster Mosaic Layer tool is executed, the tool examines the image files that are referenced by the raster catalog and checks for a shapefile of the same name in the same directory. If such a file exists, the footprint for that image file in the raster catalog will be modified to be the union of the polygon features found in the shapefile.

By using one or both of the above approaches, you can modify the footprint polygons in a raster catalog to be an accurate representation of the valid data in their associated rasters and not just the rectangular extents of the raster file. This added information stored in the raster catalog is then leveraged by the Raster Mosaic Layer during map rendering to render the desired subset of each raster, thereby achieving masking of the raster's NoData regions.

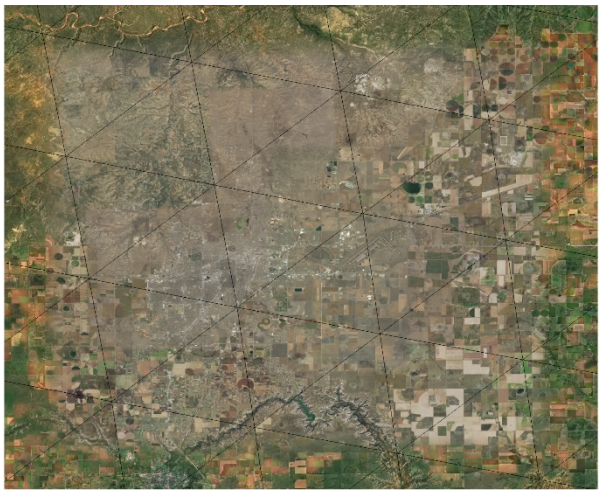
Note that these approaches can be used to mask out areas in the imagery that contain clouds or cloud shadows if you have underlying imagery that does not have these cloud artifacts. (If you do not have imagery to substitute the cloud artifacts and the affected area is small, consider removing them from the imagery in an image-editing application with a cloning tool.)

Edge blending

Using a Raster Mosaic Layer, you can feather blend the edges of a mosaic to achieve a smooth transition to the underlying dataset. Edge blending is a simple way to greatly reduce the appearance of hard borders between imagery datasets in your terrain. The transition between foreground and background imagery is achieved through a crossfade from foreground imagery to background imagery and is applied in meters. Be sure to exclude any NoData areas (as described in the previous steps of this chapter) within your mosaic before assigning an edge blend.



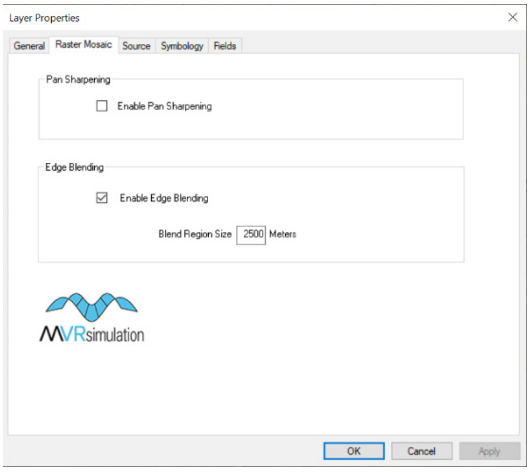
Without edge blending.



With edge blending applied.

To enable edge blending:

1. In the TOC, right-click on the RML layer and select Properties from the context menu.
2. When the Layer Properties dialog box appears, click the Raster Mosaic tab.
3. Select the Enable Edge Blending checkbox, and in the Blend Region Size field enter the size of the blend region (in meters).



4. Experiment with different values for the Blend Region Size and observe the feather blending results in the map by clicking the Apply button, which enables you to see the effects of a change in blend distance immediately, without exiting from the dialog box.
5. Click OK to exit from the dialog box.

Note: The Raster Mosaic Layer can also contain elevation rasters. If a smoother transition between elevation datasets is desired, you can apply a blend region to help smooth out the terrain between datasets.

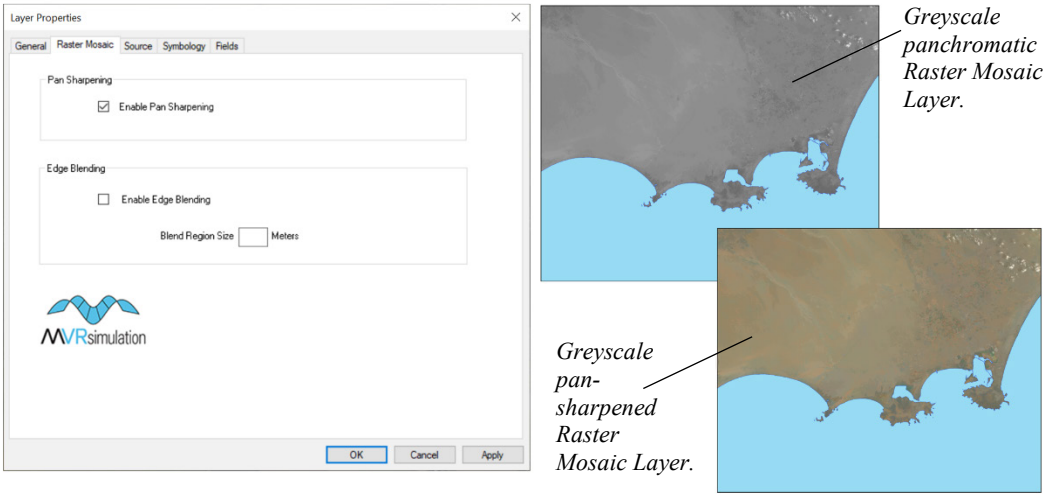
Pan sharpening

The process of pan sharpening as implemented by Terrain Tools Raster Mosaic Layer is used to *fuse* color information from an underlying color imagery layer to a 1-band panchromatic (greyscale) imagery layer. This is useful if you have a higher resolution greyscale imagery layer that you would like to give the appearance of being colorized. The pan-sharpening transformation operates on an entire mosaic layer. By fusing the panchromatic intensity information found in the greyscale imagery to an underlying color mosaic layer, enabling pan sharpening gives the greyscale imagery layer the appearance of being a colorized mosaic layer.

To enable pan sharpening:

1. In the TOC, position the Raster Mosaic Layer(s) containing panchromatic/greyscale imagery above a color mosaic layer in the imagery group.
2. Right-click on the Raster Mosaic Layer to be pan-sharpened and select Properties from the context menu.

3. When the Layer Properties dialog box appears, select the Raster Mosaic tab and select the Enable Pan Sharpening checkbox.
4. Click OK to exit from the dialog box.



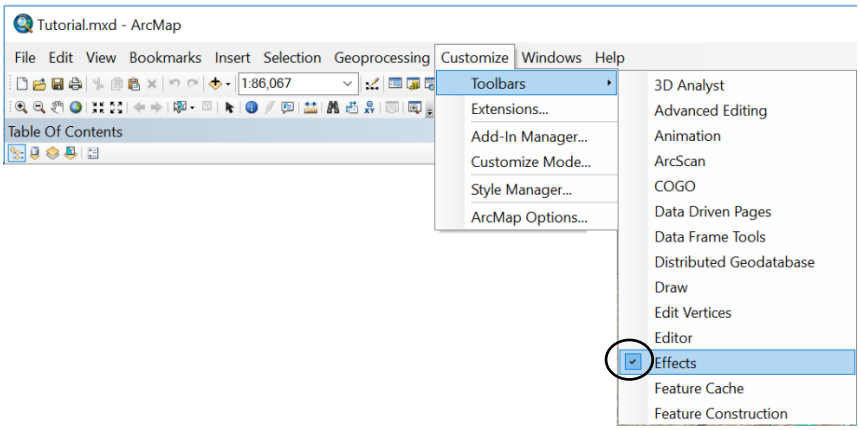
Color adjustments

Adjusting the color, contrast, and brightness of a Raster Mosaic Layer can help blend the transition between imagery layers so that they appear more seamless.

Adjusting contrast & brightness using the Effects toolbar

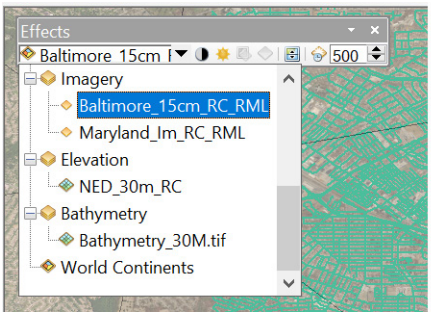
To adjust the contrast and brightness of a Raster Mosaic Layer:

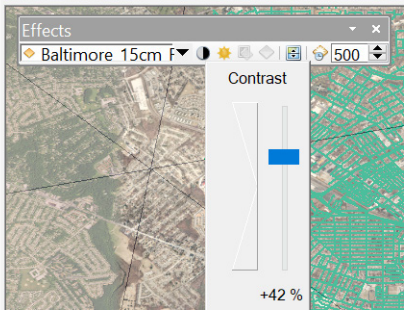
1. DisplayArcMap's Effects toolbar, if it is not already displayed, as shown:



2. On the Effects toolbar, select a Raster Mosaic Layer from the pulldown list.

3. Select the Contrast or Brightness icon to the right of the pulldown menu and adjust the contrast or brightness level by moving the slider up or down. To restore the layer's original contrast or brightness level, reset the level to "0".



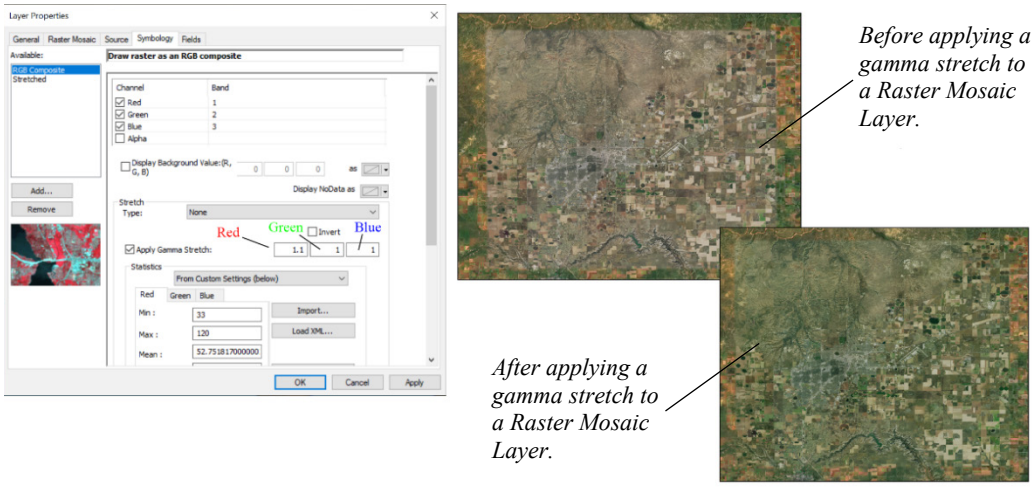


Applying a gamma stretch

By applying a gamma stretch correction to a Raster Mosaic Layer, you can control the ratios of red to green to blue to help you achieve a more seamless appearance between different imagery layers. Doing so will lead to visually smooth transitions when the imagery is compiled into 3D terrain and visualized in VRSG.

To adjust the RGB levels of a Raster Mosaic Layer:

1. On the TOC, right-click on a Raster Mosaic Layer and choose Properties from the context menu.
2. When the Layer Properties dialog box appears, select the Symbology tab and select the Apply Gamma Stretch checkbox.
3. To the right of the Apply Gamma Stretch, the three fields are the RGB values for Red, Green, and Blue, in that left-right order. These values need very little adjustment to affect the overall result. For example, to increase and brighten the red values in the mosaic, change the red value from 1 (default) to 1.1.
4. You can also subtract from a value as well. For example, to decrease the red in your mosaic layer, change the 1 to 0.90. The "0" before the decimal point is required.
5. From the Statistics drop-down list, select From Custom Settings Below. (If this option is not selected, you may see color anomalies in your terrain tile build.)
6. Click Apply to see your changes. You can continue to adjust these numbers after previewing the results. To restore the layer's original color, just type "1" in each field and click OK.
7. Click OK when finished.



LOD morphing in VRSG terrain tiles

VRSG renders the transition between levels of detail (LODs) in terrain tiles with a technique called *LOD morphing*. Introduced in Terrain Tools 1.6, terrain tiles built with LOD morphing enabled, ensure that terrain vertices are blended seamlessly between LOD transitions. This technique replaces the legacy LOD fade-blending method used in previous versions of Terrain Tools and rendered in versions of VRSG older than version 6.3.34.

The benefits of LOD morphing include rendering less geometry in the scene such that terrain with standard 60-meter post spacing will render over 22.5% fewer triangles. This transition also reduces visual artifacts between LOD transitions within mountainous regions in VRSG. If you are currently building terrain with Terrain Tools 1.6 or newer, LOD morphing is enabled by default.

Handling terrain with legacy LOD fade-blending

When loading terrain tiles in VRSG that were built with both legacy LOD fade-blending and tiles with the newer LOD morphing, the LOD transition between tilesets may produce a visible crack in VRSG v6.4 or newer. You can remedy this situation by:

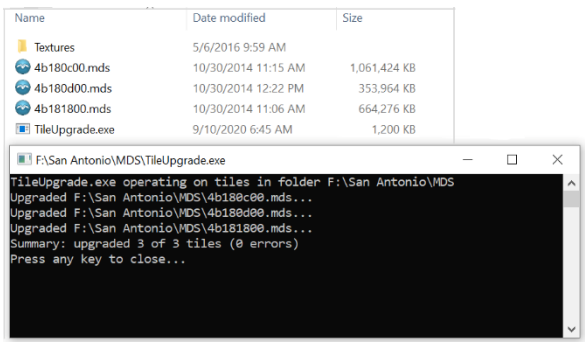
- Enabling VRSG to render all terrain with the legacy LOD fade-blending preference. You can unselect the Enable Morphing LODs checkbox on the VRSG Dashboard Graphics tab. This action will force VRSG to render all terrain with the legacy LOD fade-blending.
- Recompile the legacy terrain tiles in Terrain Tools v1.6 or newer to use LOD morphing. This preference is enabled by default.
- Use MVRsimulation's TileUpgrade utility to update the terrain automatically. (This is the simplest way to update existing terrain tiles and the process is described fully in the following section.)
- Recompile terrain to use the legacy LOD Fade-blending method. Set the Terrain Tools INI preference enableLODMorphing to false and recompile the terrain. You can add this preference in your user.ini file located in \MVRsimulation\ArcGIS Terrain Tools\Config\TerrainTools_*name*_user.ini:

```
[MVR_Config]
        enableLODMorphing = false
```

Adding or removing LOD morphing using TileUpgrade

To simplify the management of LOD morphing for legacy terrain, MVRsimulation provides a command-line utility to add or remove morphing data from a given set of terrain tiles. This utility, TileUpgrade.exe, is located in \MVRsimulation\ArcGIS Terrain Tools\Util. This utility updates legacy terrain to use LOD morphing; it can also disable LOD morphing to impose the legacy LOD fade-blending method on a set of terrain tiles.

You can simply choose to copy the TileUpgrade.exe into a directory containing *.mds terrain tiles. Once copied, double-click on the executable to begin the process of upgrading the mds tiles to use LOD morphing. An example is shown below:



The following table illustrates the syntax of the TileUpgrade.exe:

Syntax	Description
TileUpgrade.exe	Upgrades legacy tiles in the exe's directory.
TileUpgrade.exe <*path/*.mds*>	Upgrades all legacy tiles in the given path.
TileUpgrade.exe <*tile*>	Upgrades a single legacy tile.

Options:

`-revert` (removes LOD morphing from tiles)

`-force` (recomputes any existing tile LODs)

The `-revert` option converts terrain with morphing LODs to the legacy LOD fade-blending method to allow it to be rendered in older versions of VRSG.

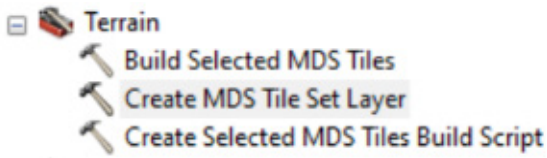
When you run TileUpgrade.exe on terrain tiles with existing morphing LODs, the tiles will be left unchanged. You can use the `-force` option to force the LODs to be recomputed.

Viewing tile coverage on a map

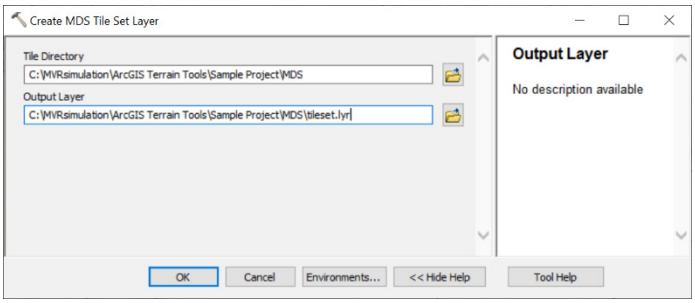
During terrain processing or once a set of terrain tiles has been built, you might find it useful to view the tile coverage on the map. You can see the coverage of a set of compiled terrain tiles by creating a tileset layer. The layer monitors a specified tile directory for changes in its contents and displays successfully completed tiles and incomplete tiles (tiles that are still being built). When any new terrain tiles are added to the directory, the new tiles are shown in the map.

To use the tileset layer for viewing the coverage of completed tiles:

1. Double-click the Create MDS Tile Set Layer tool in the Terrain toolset.



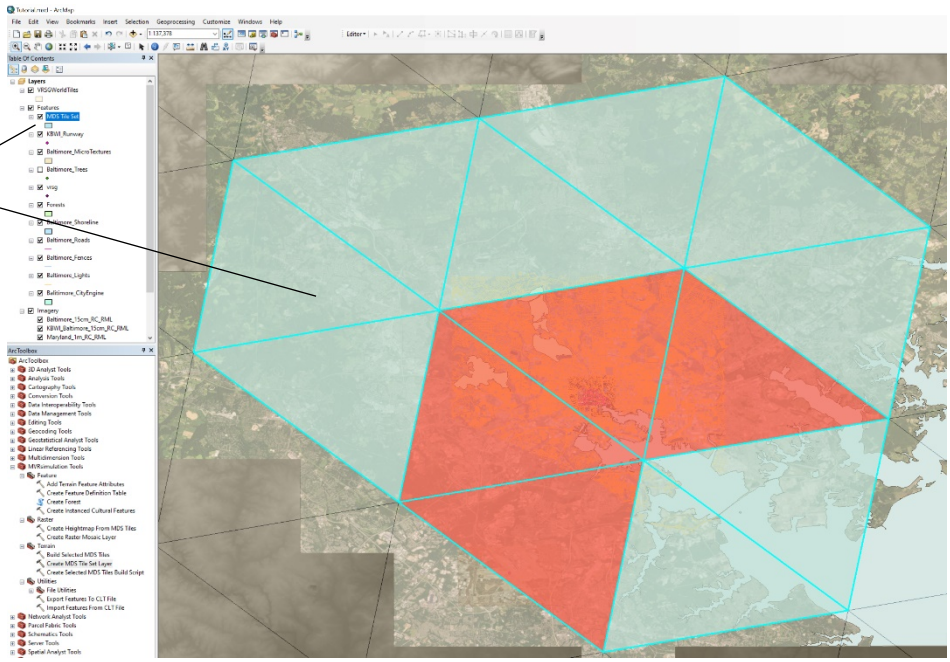
2. When the Create MDS Tile Set Layer box appears, under Tile Directory, click the browse button and navigate to the directory containing MDS terrain tiles.
3. For Output Layer, browse to output directory, and provide a name for the layer.



4. Click OK. A new tileset layer will be added to the map and listed in the TOC.

If you are currently running a terrain build, the tile completion status will be displayed on the map in the tileset layer. During terrain compilation, you will see some tiles colored in red indicating tiles are still in progress. Completed tiles will show up in the layer using the tileset layer's default symbology color.

Tile Set layer showing tile generation in progress. The completed tiles are shown in the layer color and the incomplete tiles are shown in red.



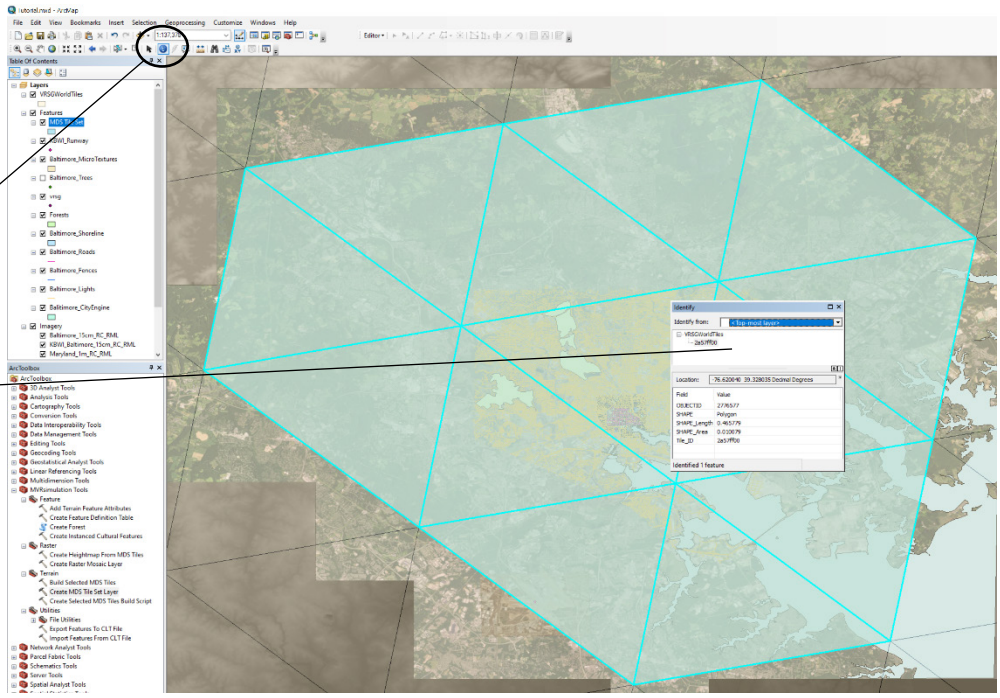
As tiles continue to build, the tileset layer should automatically update to reflect the terrain building progress. If you do not see the tiles changing colors after tiles have been complete, refresh the map.

If red tiles are present after the terrain build has been completed, the color red also indicates that those tiles were not fully built for some reason and they cannot be read. Consult the tile build log in the Terrain Tools Build Manager for error messages.

Although you can see which tiles are being built on the Build Manager window, you can also identify tiles in the tile layer on the map.

To find out the name of a tile, right-click on the tile of interest, and from the context menu choose Identify.

As shown in the example below, the Identify dialog box appears, displaying information about the tile.



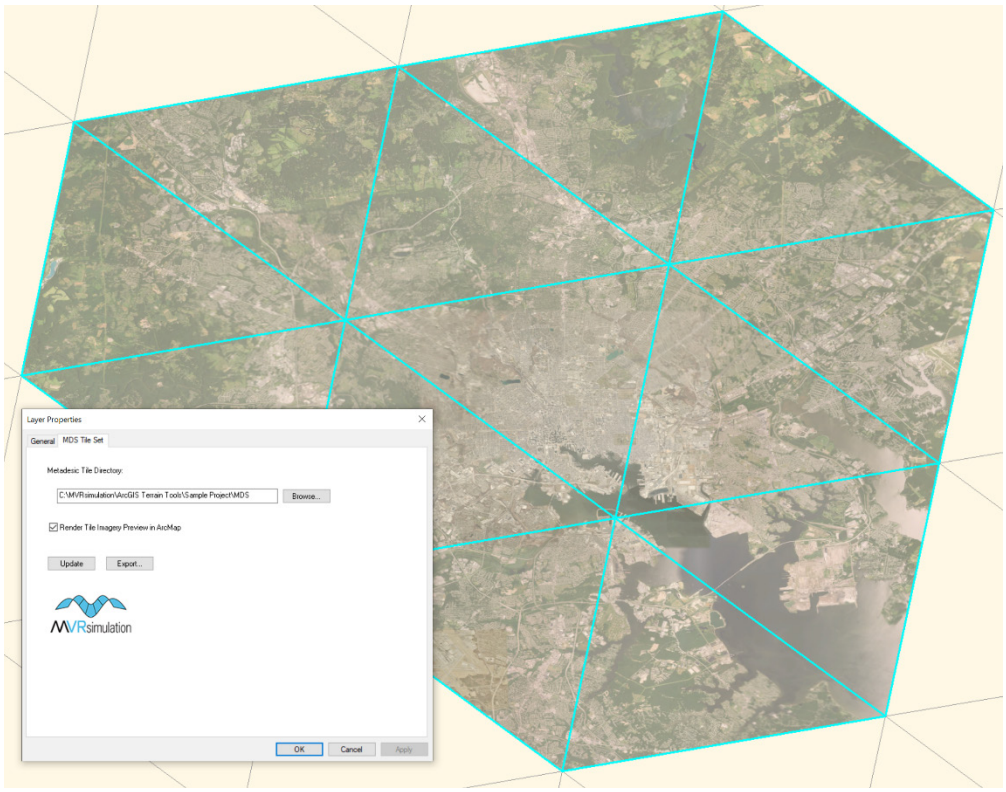
Using Tile Set Layers as source data

Sometimes you might need to modify a subset of existing terrain tiles with newer source data that you do not have access to. Up until now, you would have had to include the source imagery and elevation data used in the original terrain build to ensure imagery and elevation consistency across tiles boundaries.

New in Terrain Tools v2.0, is the ability to use a tileset layer as imagery and elevation source data. (See the previous section for information about how to create a tileset layer.) This way, you can update terrain tiles with new source data blended with previously compiled terrain tiles. This capability is automatically enabled when you place a tileset layer in the Imagery and Elevation group layers in the TOC.

Typically, you would want to put your tileset layer in the TOC under *both* Imagery and Elevation group layers to compile both imagery and elevation data used in the original terrain build. These layers should be placed at the bottom of the group order so that your new imagery or elevation source data will be compiled above these tileset layers in the TOC. For best results with blending your new source data with the existing imagery contained in the terrain tiles, create an RML layer for your new source data with Edge Blending enabled, as described earlier in this chapter. This will provide for a more seamless transition.

When you use a tileset layer as an imagery source, it can be useful to color match newer imagery with existing terrain tiles. In this case, you can optionally preview the tileset imagery in the map viewer. Note that due to rendering performance limitations, this preview works best for smaller tileset layers. To enable the tileset layer imagery preview, right-click on the tileset layer in the TOC, and on the MDS Tile Set tab of Layer Properties, select the Render Tile Imagery Preview checkbox.



Building tiles with a python script

The tutorial chapter describes how to use the Build Selected MDS Tiles tool to launch a tile building process within ArcMap. You can also save the parameters for building tiles to a script to execute at a later time. The Create Selected MDS Tiles Build Script tool enables you to save all the parameters of a tile building process to an executable script that can be run later and possibly on a different machine. (To execute the script on a different machine, that machine must have ArcGIS Desktop or ArcEngine and Terrain Tools installed). Creating and using a tile-building script is an easy way to ensure that successive tile-building processes use the same parameters. It also eliminates the effort of re-entering the values in the Build Selected Tiles dialog box during successive builds, which could be prone to error.

Although you can also rebuild tiles by using the Rebuild All function available on the Active Build page of the Build Manager (as described in the chapter, "Managing Terrain Builds"), the advantage of having a script is that it contains build parameters in a human-readable file that you can edit if necessary, or can archive in a source control system.

ArcGIS Desktop has integrated scripting capability with the Python programming language. For more information about Python or the integration of Python with ArcGIS Desktop, see the ArcGIS Desktop help.

The Create Selected MDS Tiles Build Script tool leverages this Python integration by creating an executable Python script (.py) for the output. If you are familiar with the Python

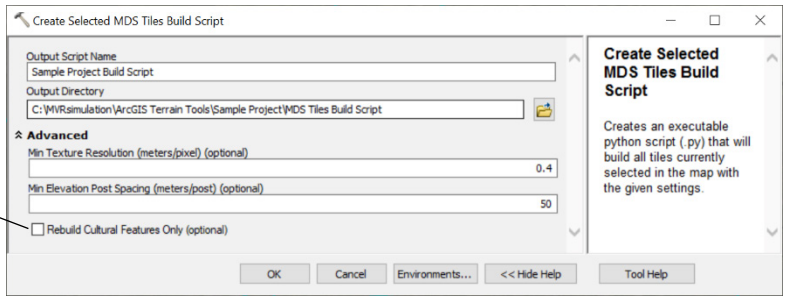
language, you could modify or extend the script that is created by MVRsimulation's tool, but doing so is generally not required.

Once the tile-building script is created, simply double-clicking the script in Windows Explorer to launches the script. The tile-building process executed by the script is independent of ArcMap; you can run it while ArcMap is open or closed. However, since the tile-building script refers to a map document (.mxd), the map document that a script uses should not be modified while the script is running.

To create a tile-building script in ArcMap:

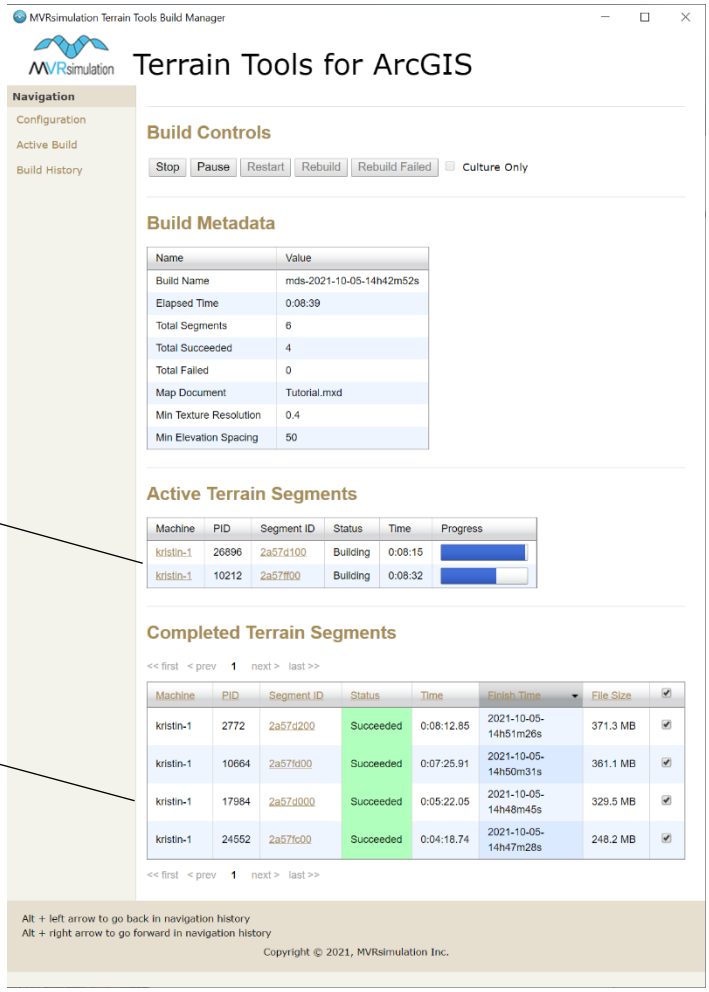
1. In the TOC, set the VRSGWorldTiles layer to be the only selectable layer and select tiles to build.
2. In the Terrain toolset, double-click the Create Selected MDS Tiles Build Script tool.

Select this checkbox to only rebuild the cultural features on an existing tile or set of tiles.



3. When the dialog box appears, enter a name for the output script name and the output directory where the script will be stored. The output directory will also be used as the destination directory for the terrain tiles when the script is executed.
4. Optionally, click Advanced, and adjust the minimum texture resolution or post spacing, in meters, for the tiles to be generated. Terrain Tools uses the resolution of the source data to determine the resolution of the compiled terrain. The minimum texture resolution and post spacing settings define a lower (finer) bound for the terrain resolution if you want the terrain to compile at a lower/courser resolution than the source imagery.
5. If you want your script to only build the culture features on the terrain, not the terrain itself, click the Rebuild Cultural Features Only checkbox. This capability will rebuild features that sit on the terrain (such as buildings, trees, fences, and lights). It will not rebuild features that cut into the terrain (such as roads, runways, or bodies of water). For more information, see the chapter, "Working with Terrain Features."
6. Click OK.
7. Click Close to close the dialog box.
8. To run the tile-building script, locate the script (.py) in Windows Explorer and double-click it.

   While the script executes in the background, the Build Manager window appears displaying the status of terrain tile processing, as shown in the following example:



Information about the current terrain building process(es).

Information about the completed terrain tiles.

When the generation of terrain tiles has been completed, all the tiles will be listed in the Completed Terrain Segments table and no remaining tiles will be listed in the Active Terrain Segments table. For more information about building terrain tiles, see the chapter, "Managing Terrain Builds."

CHAPTER 4

Working with Terrain Features

In addition to using imagery and elevation data to build 3D terrain, you can use feature data as input. In some cases, terrain features modify the basic terrain skin that is generated from the imagery and elevation source.

Terrain Tools provides polyline, polygon, and point features, which are explained in depth in this chapter.

Overview

Terrain features consist of geographical features that can be represented by points, polylines, or polygons. You can create and edit features in ArcMap by digitizing features from underlying imagery layers or by incorporating them from existing open data sources.

Terrain Tools terrain features are briefly described below.

Polyline features:

- Road features – for adding 2D roads that are cut into the terrain at a specified width.
- Wall features – for adding 2D extruded fences or walls on the terrain.
- Cultural light features – for creating compiled-in light points located along polylines at a specified height to render at night in VRSG.
- Powerlines – for creating instanced powerline models with connected wire geometry.

Polygon features:

- Forests features – for generating large geotypical forests or other vegetation or by assigning vegetation models within polygonal boundaries.
- CityEngine features – for compiling building models using an Esri CityEngine rule package (.rpk).
- Microtexture features – for adding geotypical ground textures to specific terrain areas to enhance the realism of the ground surface when it is viewed at close range.
- Water – for marking water bodies to be rendered using VRSG's 3D oceans or 2D legacy textured water.
- Terrain inset models – for incorporating inset models by fitting them into an area of cut-out terrain.

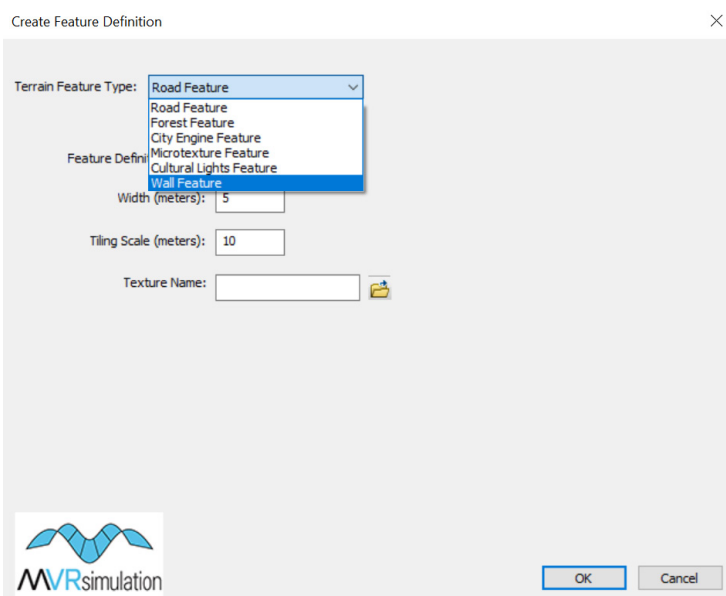
Point features:

- Runway – for incorporating a runway model into a 3D round-earth terrain tile where the model geometry is compiled into the terrain.
- Tree points – for adding tree models to geospecific tree locations.
- Other 3D modes – to compile models into the terrain tiles.

In Terrain Tools, features need to be assigned an MVR Feature Definition which will instruct the application on how to compile these features into the terrain. Some features require the user to define a Feature Definition Table to customize their attributes. These features include roads, forests, CityEngine buildings, microtextures, cultural lights, and wall features. Other feature types use existing Terrain Tools Feature Definition Tables which are already pre-populated with the necessary attribute information such as water, runways, and other terrain modification features.

Using feature definitions tables

Terrain Tools provides an editable interface for defining MVR Feature Definitions. Depending on the selected feature type, the user-defined attributes are listed within the interface allowing you to easily populate a Feature Definition Table with the necessary attribute information for each feature type.



The Create Feature Definition Table tool, located in MVRsimulation's Features toolbox, creates an empty Feature Definition Table for a point, polyline, or polygon feature. The output Feature Definition Table is created on disk within a File Geodatabase (.gdb) within the same directory as the corresponding input feature class.

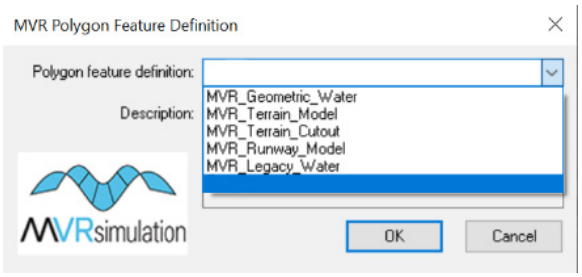
The Feature Definition Table can be accessed by using the ArcCatalog window within the ArcMap interface. Expand the contents of the File Geodatabase to see the table. Right-click on the Feature Definition Table within the ArcCatalog window to select the Create

MVRsimulation Feature Definition option from the dialog menu to bring up the Create Feature Definition box. Use this menu to populate the Feature Definition Table with an MVRsimulation Feature type from the pulldown menu.

Other feature definition types not included in the Create Feature Definitions box but are included in Terrain Tools by default are:

- MVR_GeometricWater - is a water polygon feature that supports VRSG's rendering of 3D oceans and bathymetry where the coastline is cut out of the terrain geometry at sea level.
- MVR_Legacy_Water – is a water polygon feature that renders water in VRSG with textured legacy water. This water feature should be used with lakes or rivers.
- MVR_Terrain_Model - is a point feature that compiles an inset model while preserving existing relative elevations in the model.
- MVR_Terrain_Cutout - is a polygon that marks an area to be cut out and removed from the terrain skin.
- MVR_Runway_Model - is a point feature that signifies the placement of a 2D model that conforms to the terrain geometry.

During an edit session, these built-in feature definition types are installed with Terrain Tools and are available from the MVR Simulation Feature Definition box as shown below:



After you create a user-defined Feature Definition for roads, forests, CityEngine buildings, microtextures, cultural lights, or wall features; those features will also appear in the dropdown menu pictured above with its designated feature name as defined by the user in the creation of the Feature Definition.

Each of these Features Definitions, user-defined or not, will need to be assigned to a feature layer in the TOC depending on the feature layer type during an active edit session. Each feature layer also requires a FeatureDef attribute in its attribute table before the feature definition assignment. This attribute can be added by running the Add Terrain Feature Attributes tool which is also located in the Features toolbox.

Each of these feature definitions will be explained in more detail later on in the chapter.

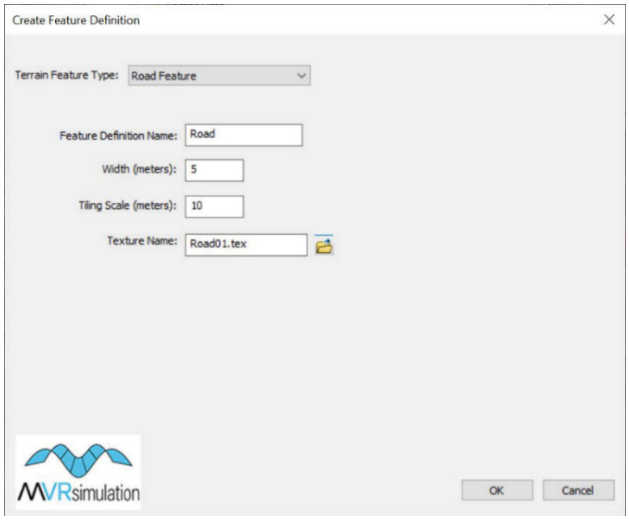
The Terrain Tools tutorial also provides step-by-step instructions using the Terrain Tools Sample Project as input for each of these feature definitions outlined above. It is strongly recommended that you complete the Sample Project before pursuing your terrain project.

For more information on how to create a Feature Definition, seesee the chapter "Building 3D Terrain Tutorial" and locate the section "Adding Terrain Features."

Creating a road feature

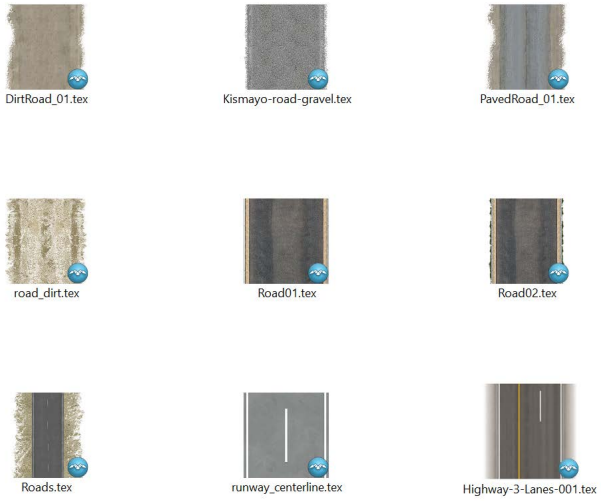
For applications such as a simulation with vehicles driving on a paved or dirt road, textured linear (polyline) features can be used for adding 2D roads that are cut into the terrain at a specified width. Road features are created using polyline feature data. You can choose to digitize roads directly from imagery or you can obtain polyline road vectors from open data sources such as Open Street Map.

To incorporate road vectors into terrain as 2D roads, a user-defined Road Feature Definition is required. A road feature is defined in a Feature Definition Table by specifying a road width, tiling scale, and road texture.



Several road Feature Definitions can be created for compiling different types of roads. For instance, you can create definitions for paved, dirt, gravel, or multilane roads using your own source textures or textures from the Terrain Tools installation directory.

MVRsimulation supplies road textures which you can use in your project and are located in the following directory: \MVRsimulation\ArcGIS Terrain Tools\Textures\Road.



You can also choose to assign your own road textures as well. When assigning a texture to a road feature definition table, the user interface only specifies .rgb and .tex file types. You can specify other texture file types by specifying the full filename with its extension.

The following file formats are supported for textures:

BMP	SGI - RGB, RGBA, INT, and INTA
GIF	TIF, TIFF
JPEG	TEX (MVRsimulation's proprietary texture format)

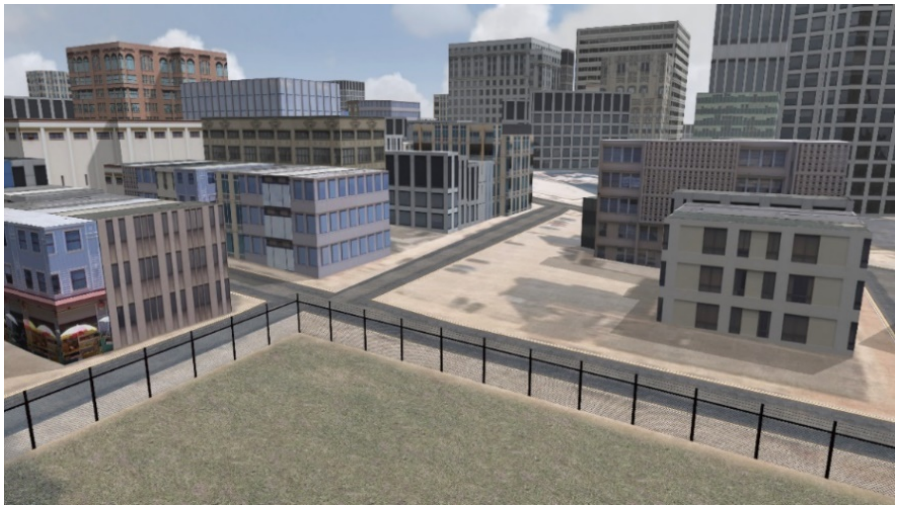
After road vectors are assigned a user-defined road Feature Definition in a FeatureDef attribute and are placed under the Features group layer in the TOC, they are ready for Terrain processing. During terrain tile generation, Terrain Tools creates a \Textures subdirectory within the terrain tiles output directory and copies any MVRsimulation-supplied texture you used to that directory. You can use your textures as well and set up a preference to point Terrain Tools to their directory location (as described in the chapter “Managing Terrain Builds”).

The Terrain Tools tutorial also provides step-by-step instructions for adding roads to the terrain using the Terrain Tools Sample Project as input. It is strongly recommended that you complete the Sample Project before pursuing your terrain project.

For more information on how to create road features, see the chapter “Building 3D Terrain Tutorial” and locate the section “Adding Roads.”

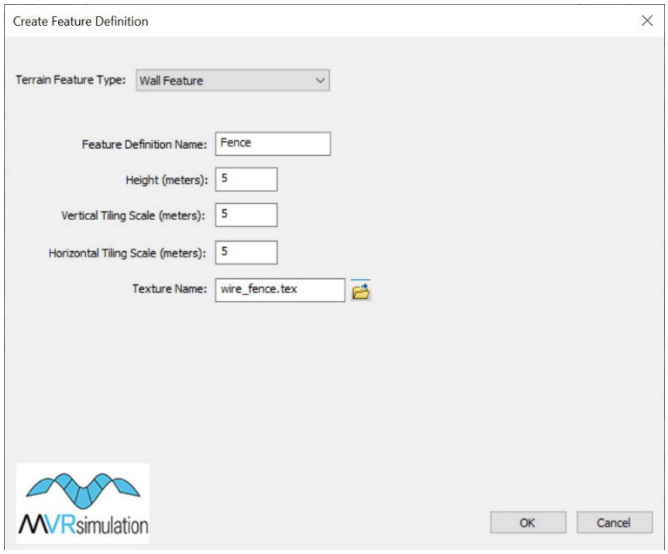
Adding a fence or wall

In Terrain Tools, the Wall polyline feature type is used for creating 2D extruded walls and fences. The following example shows a portion of a fence placed on the tutorial sample project terrain using a fence texture delivered with Terrain Tools:



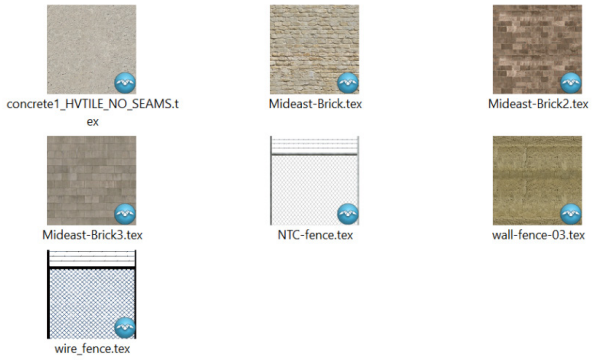
For terrain projects requiring extruded 2D fences or walls, textured linear (polyline) features can be used for adding 2D fences that are placed on the terrain at a specified height. Wall features are created using polyline feature data. You can choose to digitize fences directly from imagery or you can obtain polyline fence vectors from open data sources.

To incorporate wall vectors into terrain as 2D walls or fences, a user-defined Wall Feature Definition is required. A wall feature is defined in a Feature Definition Table by specifying a height, vertical tiling scale, horizontal tiling scale, and texture name.



Several different fences or wall Feature Definitions can be created for compiling different types of walls. For instance, you can create definitions for a wire fence or a brick wall using your own source textures or textures from the Terrain Tools installation directory.

MVRsimulation supplies wall textures which you can use in your project and are located in the following directory: \MVRsimulation\ArcGIS Terrain Tools\Textures\Wall.



After wall or fence vectors are assigned a user-defined wall Feature Definition in a FeatureDef attribute and are placed under the Features group layer in the TOC, they are ready for terrain processing. During terrain tile generation, Terrain Tools creates a \Textures subdirectory within the terrain tiles output directory and copies any MVRsimulation-supplied texture you used to that directory. You can use your textures as well and set up a preference to point Terrain Tools to their directory location (as described in the chapter “Managing Terrain Builds”).

The Terrain Tools tutorial also provides step-by-step instructions for adding fences to the terrain using the Terrain Tools Sample Project as input. It is strongly recommended that you complete the Sample Project before pursuing your terrain project.

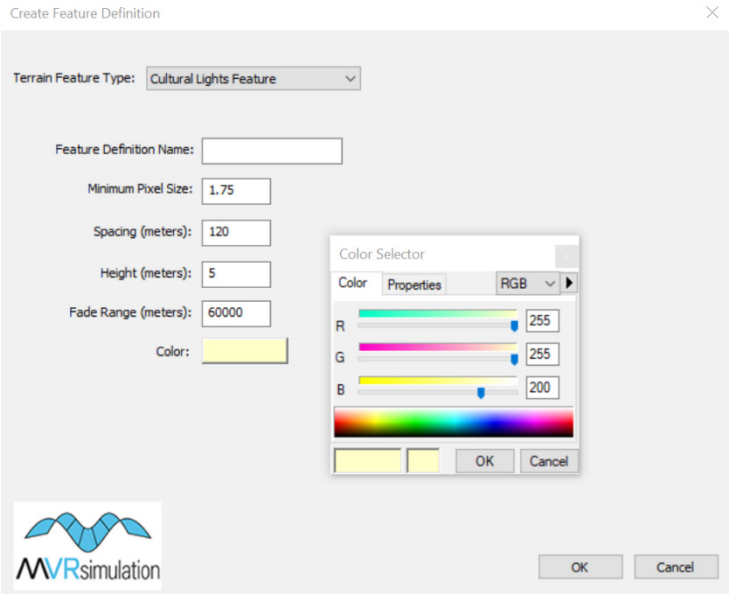
For more information on how to create wall features, see the chapter “Building 3D Terrain Tutorial” and locate the section and locate the section “Adding Extruded Fences.”

Adding cultural lights

You can distribute light points along a polyline with the Cultural Lights feature to give the appearance of streetlights when the terrain is rendered in VRSG at night.

Cultural light features are created using polyline feature data. You can choose to digitize polylines or you can simply use road vector polylines as source data for cultural light placement as streetlights are generally located along roads.

Light point attributes are defined in an MVRsimulation Cultural Lights Feature Definition.



To incorporate cultural lights into the terrain as light points, a user-defined Cultural Lights Feature Definition is required. A Cultural Lights feature is defined in a Feature Definition Table by specifying a color, pixel size, height, fade range, and spacing.

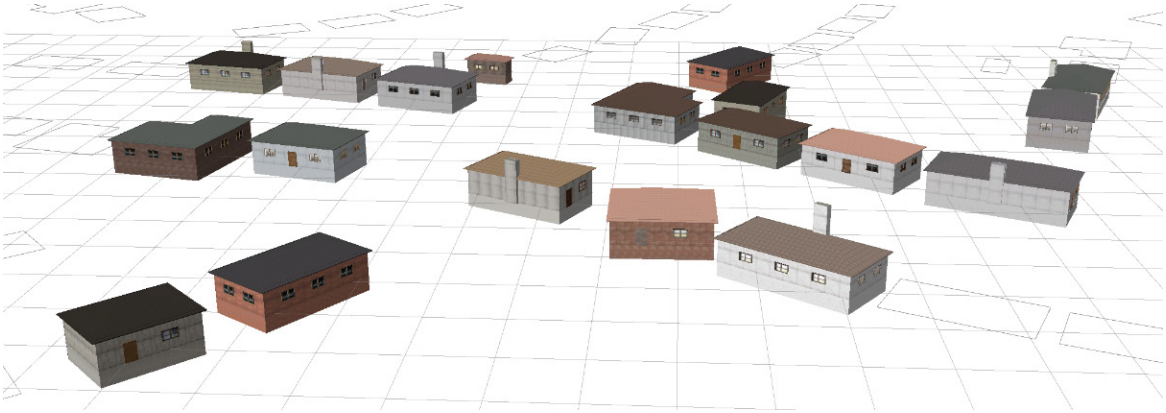
After defining the attributes for the light points in a user-defined Cultural Lights Feature Definition, assign the Feature Definition to the FeatureDef attribute in the attribute table. Upon building terrain, place the Cultural Lights feature under the Features layer in the TOC. Terrain Tools will compile the light points directly into the terrain tiles based on the user-defined light point attributes.

Terrain Tiles with compiled-in light points can be rendered in VRSG by adjusting the Time of Day on the Environment tab on the VRSG Dashboard. (Light points will not render in VRSG during the daytime.)The Terrain Tools tutorial also provides step-by-step instructions for adding cultural lights to the terrain using the Terrain Tools Sample Project as input. It is strongly recommended that you complete the Sample Project before pursuing your terrain project.

For step-by-step information about how to add light points to the terrain, see the chapter "Building 3D Terrain Tutorial."

Adding buildings from a CityEngine rule package

Terrain Tools contains a feature type for generating buildings using an Esri CityEngine rule package (.rpk). This feature type replaces the legacy way of making extruded buildings in Terrain Tools version 1.5 and earlier. Using an RPK within Terrain Tools does not require the use of CityEngine.



An RPK is a compressed 7-Zip archive package, created in Esri's CityEngine software product (not part of ArcGIS). An RPK contains compiled Computer Generated Architecture (CGA) rule files (*.cgb), and all referenced assets and data. RPKs can be applied to any building footprints.

The texturing and geometric complexity is defined by the compiled CGA rule files. This can vary depending on the end-use case. Typically, the procedurally made building models generated from an RPK are meant to cover a footprint that can be seen on the 3D terrain at a distance, such as in-flight simulations. The buildings tend to be less detailed than those created manually by 3D artists. However, the ability to cover huge areas procedurally is compelling. Additionally, for increased variation and customization, fields in the feature class can be mapped to variables in the .cga automatically. For instance, a .cga rule might contain a BuildingHeight attribute for setting the height of the extruded building. If the input feature class has a matching BuildingHeight field containing building height data from GIS sources, the tool will automatically map the feature attribute value for a given building.

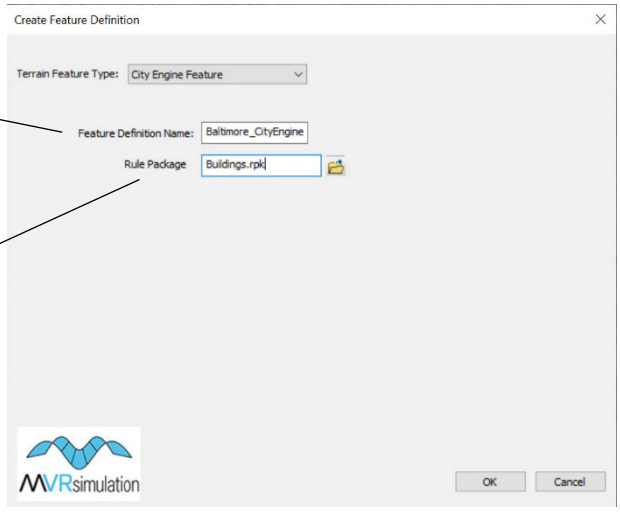
If you have CityEngine, you can make your rule packages or modify existing ones to suit your needs. (See https://doc.arcgis.com/en/cityengine/2019.0/help/help-rule-package.htm for creating your rule package.) Rule packages are also publicly available from Esri, at: https://www.arcgis.com/home/group.html?id=023a5bfcb98f467f9c1673423ad75fd1#overview:

Using an existing RPK means that you are forced into using its rules. For example, if you are working on terrain that requires buildings appropriate for a tropical area but you have an RPK for Paris buildings, you should find an available RPK more suited to your terrain or obtain CityEngine and make your own rule package. You can also modify an existing rule package from within CityEngine.

RPKs that have been curated and used by MVRsimulation are available for customer use. You can download them from the company's Download Server in the section Software/Terrain Tools/CityEngine Rule Packages (RPKs) by MVR.

Name of the feature definition.

Browse for the RPK you want to use.



If you need to use high-resolution detailed models from MVRsimulation's robust library of building models or other models at your own site (in MVRsimulation's model format), consider using MVRsimulation's Scenario Editor after you build your 3D terrain. In Scenario Editor, you can populate the 3D terrain with lots of high-resolution cultural features of the kind used in many of MVRsimulation's 3D terrains available to customers. Scenario Editor has a game-level interface, in which you place model instances on the terrain in a WYSIWYG manner.

Some MVRsimulation customers use a mix of geotypical lower-resolution buildings from CityEngine with higher resolution geospecific models of buildings needed as landmarks or targets.

Other MVRsimulation customers use CityEngine to generate a 3D model of a large number of parameterized geotypical buildings of an urban area. The resulting model can be placed externally on the compiled terrain via the terrain's cultural feature file (vrsg.clt), or compiled into the terrain using Terrain Tools. The latter uses the same feature class format as instancing features. For more information about this workflow, see the appendix "Adding CityEngine Generated Models to 3D Terrain."

Optimizing City Engine output from CGA rules with LODs

The procedural output of CityEngine often results in a large area with dense geometry. If you are authoring your own .cga rules, make sure to optimize the output. The following are two main approaches you can leverage.

First, for repeated geometry that does not depend on the footprint shape, such as doors, windows, and rooftop elements, you can factor out that geometry into an external model. Then, reference the external model using the CGA insert operation in the .cga rule. Terrain Tools will automatically cluster and serialize this data as instanced geometry, to optimize rendering performance. For example, this rule inserts an external window model in .obj format:

```
Window -->
    case LOD == "high":
      setupProjection(0, scope.xy, 512, 512)
      projectUV(0)
      i(WindowModels, yUp,alignSizeAndPosition)
      s('1, '1, '1)
      t(0,0,-.079)
      texture(fileRandom(LedgeTextures))
```

Additionally, as shown in the example, Terrain Tools supports scripting levels of detail (LODs) in CGA rules. This support is implemented as an enum annotation, which defines the LOD states, and an attribute that stores the active LOD. This LOD attribute is then used to conditionally simplify the output for lower detail LODs. The LOD states in the enumeration are ordered from highest to the lowest detail. For example, at the top of your CGA rule, you could add the following:

```
@Group ("Level of Detail",0) @Order(0) @Enum("high", "medium", "low")
attr LOD = "high"
```

This LOD attribute is then referenced in the rule file when generating the different parts of the shape geometry. The following example defines the building wall texturing. The low LOD has no windows or doors, just a single face with a wall texture, whereas the medium and high LODs partition the wall surface into areas of walls, windows, and doors:

```
Texture(whichFace) -->
   case whichFace == "Wall":
      case LOD == "low":
          setupProjection(0, scope.xy,1,1)
          projectUV(0)
          texture(WallTexture)
      else:
          setupProjection(0, scope.xy,1,1)
          projectUV(0)
          texture(WallTexture)
          SplitWalls   // rule splits the wall into door/window/wall
```

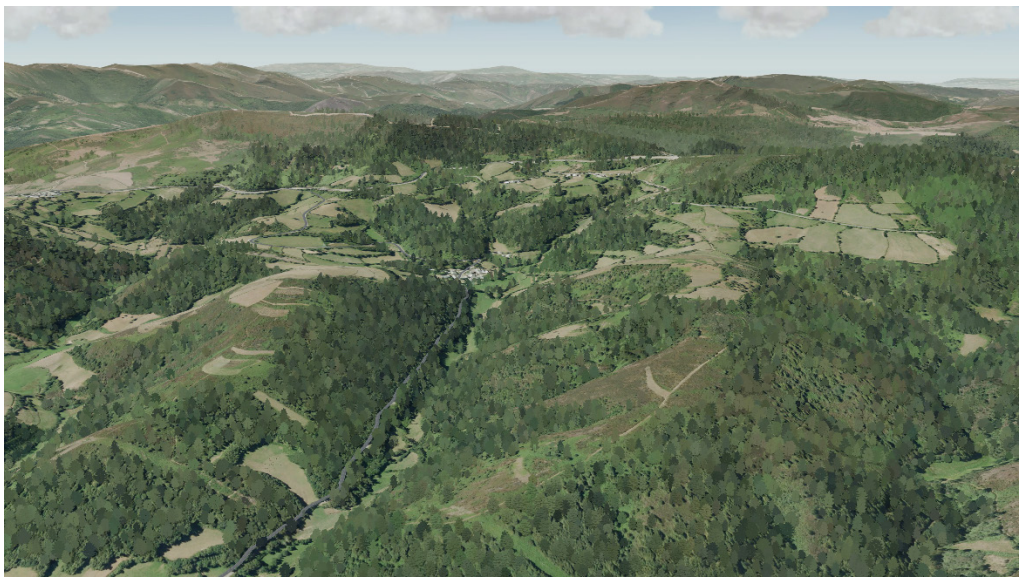
The enumeration of LOD states can be arbitrary strings, or even integer or floating-point values that define the maximum range for that LOD. This example used strings with well-known values of “high”, ”medium” and “low”. If numerical values are used to define the LOD states, these values will be used as the maximum camera range that the buildings for that LOD will be visible from. If string values are used the LOD ranges are derived from an INI preference `defaultCityEngineLODRanges`. The value of this preference is a comma-separated list of integer or floating-point numbers. These numbers are the maximum range a given LOD is visible from, ordered from the highest detail to the lowest. The vanishing distance is always the last value in the list. The default values are:

```
defaultCityEngineLODRanges = 1500, 3000, 5000, 7500, 10000
```

This preference describes the geometry generated from the “high” LOD state as visible from 0-1500 meters, the “medium” geometry as visible from 1500-3000, and the “low” geometry as visible from 3000-10000 meters.

Adding forests and large areas of vegetation

There are a couple of ways to generate a dense coverage of vegetation, such as forests or other types of ground cover.



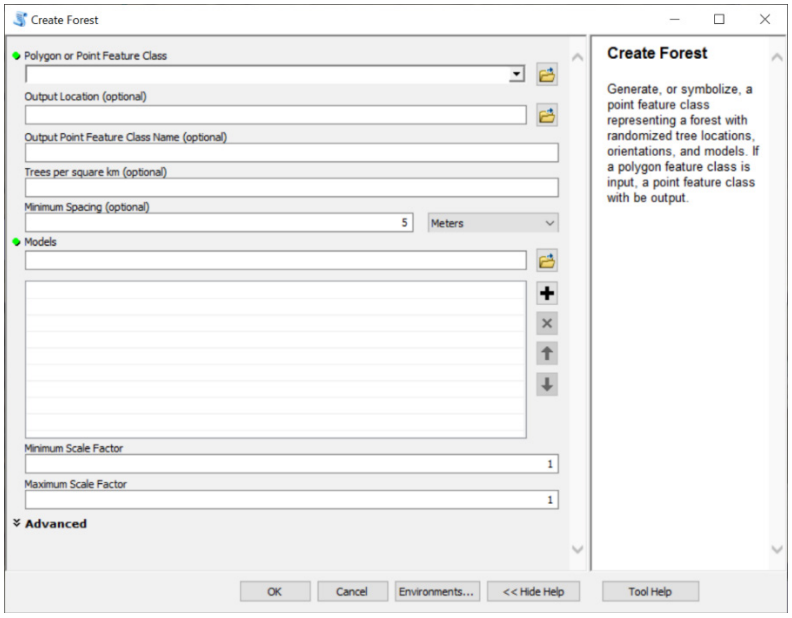
Example in VRSG of using the Create Forest tool to populate a heavily forested area on 3D terrain of Lugo, Spain. Notice how the trees blend well with the terrain; the tree colors are sampled from the terrain imagery.

You can choose to use either the Create Forest Tool or the Forest Feature Definition workflow depending on your specific needs.

Using the Create Forest tool

The Create Forest tool provides a way to easily distribute thousands of scaled model instances using a polygon or point feature class. You can specify a polygon or point feature class, assign one or more tree (or vegetation) models, randomize orientation, and then specify the scale factor. If using a polygon, you can also specify the distribution density and minimum spacing between the trees. This tool can be used for any model such as grass, bushes, rocks, and so on. Some of the benefits of using this tool include:

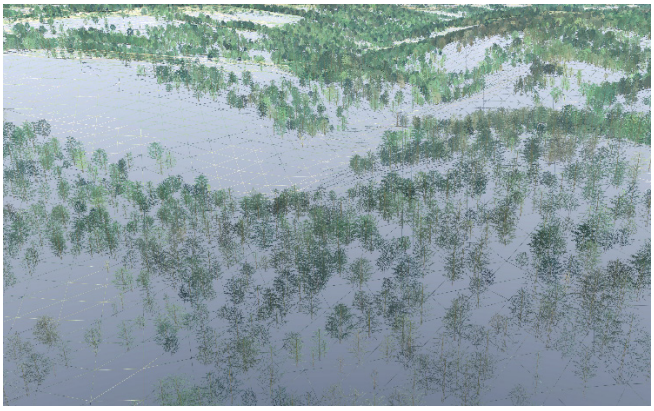
- The Create Forest tool can be used to modify existing point feature data, such as to further randomize orientation, adjust the scale, and change the model type during the editing process.
- This workflow, in contrast to creating a Forest feature workflow which is described later on, also enables you to see where tree points will be located on the terrain in the map viewer which you can select and edit as necessary.
- Tree points that are generated from Lidar data can also leverage this tool for quickly assigning tree models at tree point locations.



The Create Forest tool is included in the MVRsimulation Tools toolset in the Features toolbox. The input feature types are polygon or point feature classes. The parameters of the tool enable you to generate a point feature class populated with the necessary attributes for generating a forest or any other vegetation type.

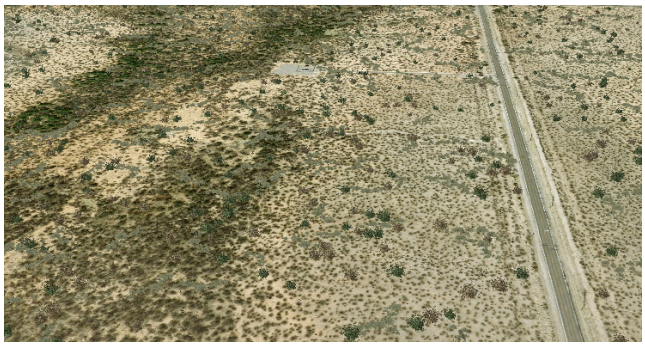
Once a point feature is generated, you can optionally choose to export these features to a .clt file, although, it is generally recommended to compile large amounts of vegetation directly into the terrain tiles by placing this file under the Features group layer during tile compilation. Compiling the trees into the terrain also enables you to take advantage of the colorizing vegetation capability from underlying imagery layers which is described later in this section.

Example in VRSG of using the Create Forest tool to populate a heavily forested area on 3D terrain of Lugo, Spain, with thousands of tree models.

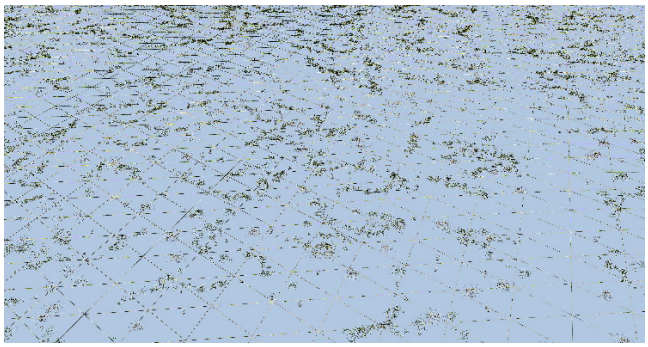


VRSG's wireframe view of the scene on the left.

This next example shows the use of the tool for populating an area with desert vegetation.

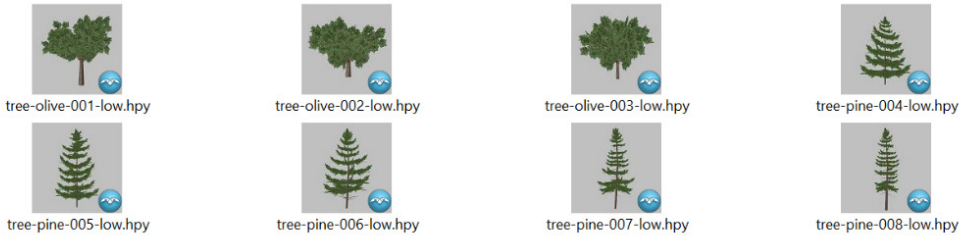


Example in VRSG of using the Create Forest tool to populate an area with multiple desert vegetation models.



VRSG's wireframe view of the scene on the left.

Tree models in MVRsimulation's model libraries that have been optimized for this workflow contain "low" in their names to indicate low polygon versions of their high-resolution counterparts and are shown below:



The tree models listed above have been optimized so that they render more efficiently on the terrain when used in large quantities.

The following table lists the number of triangles that comprise each model at each LOD level. For example, tree-olive-001-low.hpy has 276 triangles at LOD1 level, 116 triangles at LOD 2 level, and so on.

Tree model	LOD 1	LOD 2	LOD 3	LOD 4
tree-olive-001-low.hpy	276	116	16	0
tree-olive-002-low.hpy	414	163	16	0
tree-olive-003-low.hpy	436	228	16	0
tree-pine-004-low.hpy	290	12	0	n/a
tree-pine-005-low.hpy	330	12	0	n/a
tree-pine-006-low.hpy	330	12	0	n/a
tree-pine-007-low.hpy	518	18	0	n/a
tree-pine-008-low.hpy	466	18	0	n/a

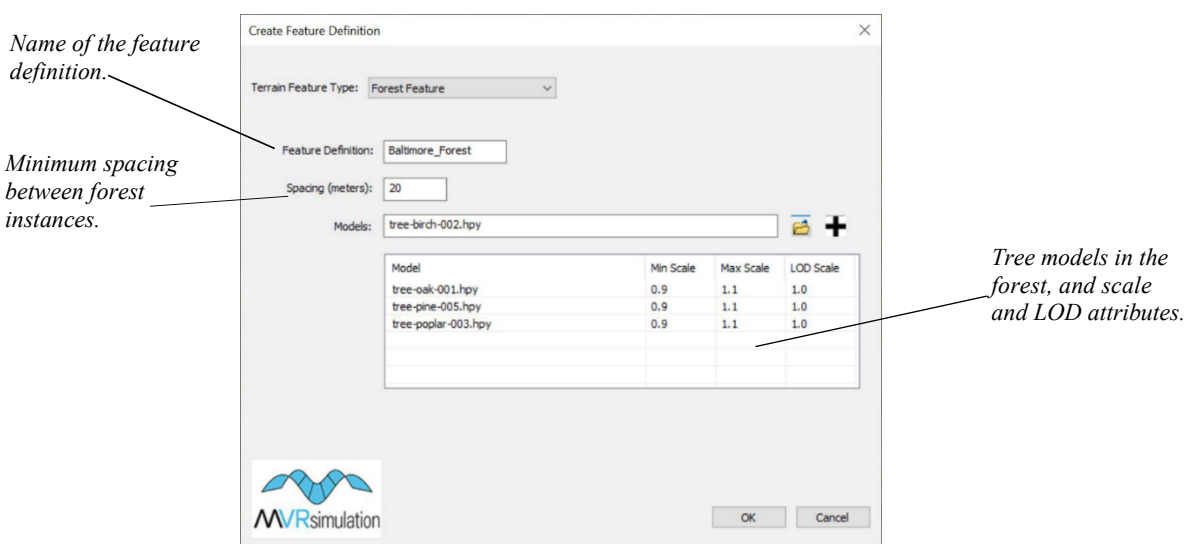
You can use this table as a general guide when you assign vegetation models to large forest features. You can view the triangle count of any tree model by opening the model in Model Viewer. The on-screen text displays the triangle count in yellow font below the model. Cycle through the models LODs by pressing the L key to cycle through the different LOD levels. (See the *MVRsimulation VRSG User's Guide* for more information about using Model Viewer.) Be mindful of a model's triangle count when creating dense vegetation features. Try to select models with optimal triangle counts and LOD transitions.

The Forest feature definition

Another option for generating a forest is to create a Forest feature within a feature definition table. New in Terrain Tools v2.0, the Forest feature definition workflow introduces a new way to populate a polygon feature with randomized vegetation models. The input for this option is a polygon. The Forest feature definition that you create encodes the various forest parameters within a Feature Definition table, and the tree positions are generated at compile time. There are a couple of advantages to using this approach over generating point features using the Create Forest Tool:

- The model instances are optimized during compilation, rather than at VRSG load time, so the terrain tiles load very quickly in VRSG.
- Additionally, VRSG can leverage its terrain paging capability to manage the model instances. The only drawback to this approach is that to reposition a compiled tree, you would have to edit the polygon feature to omit the area and recompile the terrain tile on which the tree is located. This drawback will be addressed in a future release.
- A polygon layer is much faster to render in an ArcMap map document than a point layer consisting of thousands of points. (You can, however, uncheck a point layer and enable it before compiling terrain tiles for faster layer draw times.)

The Forest Feature dialog box and its parameters are outlined below:



Colorizing tree models

Terrain Tools can colorize models of trees and other vegetation by automatically color matching the models with the underlying geospecific imagery during a terrain build. The result of this automatic sampling yields increased realism over a large area. Colorization is controlled by a preference called `sampleInstanceColor` which you can set to `true` in the TerrainTools_<machine-name>_user.ini file.

By default, vegetation models will preserve their natural color when compiled. However, you can have the underlying imagery color influence the color of model instances. This feature can offer compelling variation while making LOD transitions significantly less noticeable.

Because this feature globally changes the color of the model, it works best for vegetation models that don't combine wood and leaves into a single mesh. For example, a model could have the trunk of a tree in one mesh, and the leaves or branches in another mesh that is configured to use MVRsimulation's tree shader. That way, the colorization can be applied to the leaves and branches, not to the trunk. Many of MVRsimulation's tree models are set up this way.

To turn on automatic colorization of vegetation models, you must enable a preference that controls it in your \MVRsimulation\ArcGIS Terrain Tools\Config\TerrainTools_<machine-name>_user.ini file. (This file can be edited in Notepad or another text editor.) In addition, there are two other preferences related to this feature.

- sampleInstanceColor – Enables colorization. The default value is false. To turn on colorization, set it to "true": `sampleInstanceColor = true`
- sampleInstanceColorTol – The maximum allowable Euclidean difference in color space between a pre-calculated average vegetation color and the sampled terrain color. The default value is 250. This preference can be used to filter out instances that would produce unnatural tree shades, such as a tree placed on an area of asphalt or a rock in the imagery. Increase this number if you would prefer a greater sampling tolerance.
- overrideInstanceColorARGB – An ARGB hex color string. The RGB color defines a color with which to override the sampled color. The alpha channel defines the blend weight between the sampled terrain color and the override color. This preference is disabled by default.

Below is an example of all three vegetation colorization preferences (described above) added to a TerrainTools_*name*_user.ini file:

```
[MVR_Config]
     sampleInstanceColor = true
     sampleInstanceColorTol = 800
     overrideInstanceColorRGBA = 51, 102, 0, 50
```

Improving vegetation performance using a JSON file

Terrain Tools supports JSON files for defining model attributes, such as the ability to set probability weights for a tree model's LODs in the model's associated JSON metadata file.

This support is in VRSG version 6.4 and newer, and offers the ability to add attributes, or metadata, to 3D models, stored in a human-readable/editable file format known as JSON. You can edit JSON files directly with a text editor such as Notepad.

For a given tree model, create a text file with the extension ".json" and add three lines as shown in the example below for the model tree-euro-beech-001:

```
{
  "content": "vrsg-model-metadata",
  "version": 1,
  "lodWeights": [1,3,6,1]
}
```

Each element in the `lodWeights` entry is the probability associated with each LOD (finest to coarsest) that it will be rendered by VRSG as the finest LOD for a given instance. In the example above, "6" denotes a 6 in 11 chance that the third finest LOD in the source model will be the finest for an instance, where 11 is the sum of the weights.

JSON files must be placed in the models directory next to the tree models they describe. Terrain Tools checks for the existence of a JSON file for a given tree model (HPY or HPX) with the same model name, in the same folder, with the ".json" extension during terrain compilation, and references the JSON file when compiling the tree instances into the tiles. As well, VRSG will load and parse the model's metadata JSON file at runtime.

For more about model attributes in JSON files, see the section "Creating light lobes and powerlines for streetlight or utility pole models" later in this chapter, and the *MVRsimulation VRSG User's Guide*.

Adding ground textures

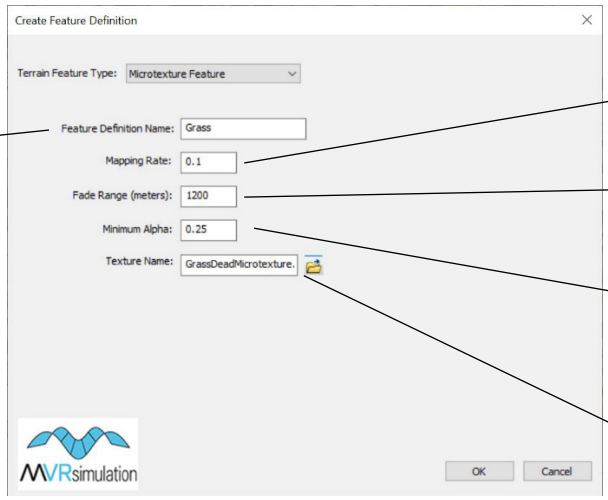
Detailed terrain textures, or *microtextures*, are high-resolution, geotypical textures that blend with the geospecific terrain imagery. For simulations that require participants to view the terrain at close range, microtextures add further detail to the terrain when the eyepoint is close to the ground. Generic textures for areas of grass, dirt, concrete, and gravel are examples of geotypical microtextures you can use in Terrain Tools to produce a realistic visual depiction of the virtual world up close.

You can apply microtextures to a set of terrain tiles in two ways:

- Provide a single overall microtexture in a directory of terrain tiles that VRSG will render in the 3D scene at runtime.
- Use a Microtexture Feature polygon to denote certain areas on the terrain where you want to assign a specific microtexture to be compiled into the terrain.

The microtexture terrain feature type in Terrain Tools can be used for depicting ground textures for just a specific area (such as a parking lot, helicopter pad, or a grassy area). Microtextures can be in any image format that VRSG supports, such as JPG, BMP, TIF, GIF, TEX, and RGB. You can add any number of microtexture features to the terrain.

In the Create Feature Definition dialog box, the Microtexture feature type attributes are shown below:



Name of microtexture. Provide a descriptive name to identify the microtexture.

Tiling rate at which the microtexture texture should map before repeating.

Distance at which the microtexture disappears completely.

Minimum percentage of base imagery texture that bleeds through the microtexture at close range.

Texture file to use for the ground texture.

The mapping frequency is calculated in units of texture repetitions per meter. A value of 0.1 means that the microtexture pattern covers an area of 10 x 10 meters.

Note: For visualizing the microtexture on the terrain in VRSG, be sure to place the texture you assigned to be a microtexture in the \Textures subdirectory of the terrain tiles directory.

Consider the following example in VRSG of a grass microtexture, shown abutting a road texture:

A VRSG close-up detail of the area, where the grass microtexture becomes visible.

Terrain Tools supports the blending of adjacent microtextures. The default blend distance is 5 meters, but you can add an attribute named “DetailBlendDist” to the feature class to increase or decrease this distance. The blending occurs automatically at the boundary that the two microtextures share so that the transition is seamless in the resulting terrain as shown in the following example of blended grass, soil, and gravel microtextures:

A VRSG close-up of an area where multiple adjacent microtextures are blended at the modeled CTS 25 MOUT site on MVRsimulation's Camp Pendleton virtual terrain.

For the blended transition to work properly, the shared boundary *must be identical in both microtextures*. The vertices along a shared boundary must meet. If the two edges do not meet exactly, the shared edge will show on the terrain as a hard edge.

The next example is a detail of the scene above, in which you can see the translucent blend areas between the adjacent microtextures:

A translucent blend area between adjacent microtextures is rendered in VRSG.

Adding runway models

Textured runways can be created as 2D models in third-party software tools, converted to MVRsimulation's HPX model format (using MVRsimulation's FBX or OpenFlight conversion utilities), and then compiled into a terrain tile. To incorporate a runway model into a 3D round-earth terrain tile, the model geometry is draped onto the terrain using the source elevation data. The terrain of the runway area is cut out and removed, and the projected model is inserted in the resulting hole. Additionally, the model geometry is modified to allow VRSG to treat it as terrain. This workflow makes the model conform to the terrain.

Before converting an Openflight runway model into an HPX model for use in Terrain Tools, be sure to add the @mvr:runway comment to any group node containing overlapping geometry such as runway stripes or taxiway lines. Any group node in the model's hierarchy containing the base geometry such as concrete should be left unchanged. This attribute should prevent z-fighting of any overlapping geometry when compiled into a terrain tile.

A runway model is represented in TOC as a point feature class and should be placed in the Features group layer. The point location of the feature is the geolocation of the origin of the model. The elevation of the point can either be represented as a Z-enabled feature, or it can be stored in a feature attribute "Elev". If this point is imported from a vrsg.clt file, the Elev attribute will automatically be populated. The supported feature attributes are similar to those supported by instanced models:

- Model <text> – the MVRsimulation HPX model containing the runway geometry.
- Elev <float> – the Z coordinate of the geolocated origin of the model.
- Yaw <float> – orientation of the model about the z-axis, in degrees (optional).
- UTM_Model <long> – a value of '1' indicates that the model was digitized against imagery in a UTM projection. It will be warped when projecting to round earth coordinates to account for this (optional). A value of '0' indicates the model did not originate from UTM data.

If you use the tool Import Features From a CLT File to import the geolocated model into ArcMap, all the feature attribute fields will be populated except for the FeatureDef attribute. This attribute is necessary to indicate that the model will be treated as a 2D terrain model. To add this attribute, you can use the Add Terrain Attributes tool to add an empty FeatureDef attribute to the point feature class attribute table. Next, assign the following feature definition: MVR_Runway_Model.

After populating the necessary attribute information, the model is ready to be compiled into the terrain when placed under the Features group layer in the TOC. When the terrain build is completed, all model geometry will be contained in the terrain tile. The textures of the runway are output to a subdirectory called \Textures within the terrain output directory, similar to other terrain features that use external textures. If you move a terrain tile to another directory, be sure to also copy or move this \Textures directory with it.

Refining runway models

Depending on the input model and the elevation source data, you might need to further refine the runway area. For example, the runway might need to be flattened using one or more areal elevation features. Additionally, runway models often will have multiple layers of geometry

(for instance, a base geometry layer for the concrete and a second layer for markings and transparent meshes).

Because vertices in both meshes are fit to the same source elevation data, Terrain Tools attempts to make the meshes geometrically conformant. By default, all layers will be cut using the terrain mesh as a lattice. This will uniformly introduce vertices to the runway, so the meshes do not intersect each other. In some cases, the terrain mesh might not be of a high enough resolution to prevent the layers from intersecting (z-fighting). To resolve this problem, you can add two Terrain Tools preferences to the \MVRsimulation\ArcGIS Terrain Tools\ConfigTerrainTools_ <machine-name>_user.ini file. (This file can be edited in Notepad or another text editor.)

The preferences are:

- modelSubdivideElevRes – the lattice mesh used to cut the runway meshes will be recursively subdivided until the space between vertices (in meters) is less than this value. The minimum value is 10 meters, to prevent introducing too much geometry.
  For example: `modelSubdivideElevRes = 40.0`
- modelSubfaceOffset – offsets transparent meshes along the z-axis by a given amount (in meters). The default offset is 5 centimeters.
  For example: `modelSubfaceOffset = 0.05`

If the runway is unacceptably bumpy, or you see z-fighting between runway layers, you might need to iterate on the runway placement and elevation of the terrain under it by visualizing it in VRSG, making adjustments to these preferences or the elevation features, and then rebuilding the affected tile.

Adding 3D terrain inset models

Terrain Tools can compile 3D terrain inset models into terrain tiles by setting a FeatureDef attribute within a polygon feature to use the MVRsimulation_Terrain_Model feature definition.

To add a 3D terrain inset model to the terrain:

1. Create or import a footprint polygon feature class containing the extents of your inset model, and add it to the Features group layer in the TOC.
2. From the MVRsimulation Tools toolbox, expand the Features toolset and double-click on Add Terrain Feature Attributes.
3. When the Add Terrain Feature Attributes dialog box appears, click the down-arrow in the Input Feature Classes field, select your 3D inset model feature class, and then click OK.
4. Right-click on your 3D inset model feature class again, and select Edit Features >Start Editing.
5. Right-click on the layer and click Selection >Select All.
6. Right-click anywhere on the map and select MVRsimulation Feature Definition.

7. When the MVRsimulation Polygon Feature Definition dialog box appears, in the Polygon Feature field, click the down-arrow, select MVRsimulation_Terrain_Model from the list, and then click OK.
8. Right-click on the layer, and select Open Attribute Table. The Model, Yaw, Elev, and UTM_Model attributes are additional variables to the 3D inset model processing.

At this point, the 3D inset model is ready to be compiled into the terrain. When the terrain build is completed, all the 3D inset model geometry will be contained in the terrain tile(s). During compilation, the textures of the inset model are output to a subdirectory called \Textures within the output terrain directory, similar to other terrain features that use external textures. If you move the terrain tile(s) to another directory, be sure to also copy or move this \Textures directory with it.

VRSG screenshot demonstrating some details of a 3D inset model -- MVRsimulation's virtual Leschi Town urban training site. The roads, curbs, bridge, and culvert are seamlessly cut into the overall terrain of Joint Base Lewis McChord, with consistent lighting and shadows.

Referencing custom or third-party models

Some terrain features use 3D model files as input, such as instancing, runways/insets, and compiled-in models. If VRSG and the MVRsimulation 3D model libraries are installed on the compilation machine, Terrain Tools automatically adds the entire 3D model libraries to its search path. However, for models that are not in MVRsimulation's 3D model libraries, you must direct Terrain Tools to where to find a given model during terrain compilation by one of these methods:

- In your feature layer's Model attribute field - Edit the Model Attribute to define a path to the directory containing models. This can be an absolute path or a path relative to the feature class location. (See the chapter "Building 3D Terrain Tutorial" for more information about the use of relative paths.)
- In your Terrain Tools user.ini file - You can define the model search path using the .ini preference modelSearchPath in your Terrain Tools User ini file. An example is shown below:

```
[MVR_Config]
modelSearchPath = "C:\MVRsimulation\ArcGIS Terrain Tools\Sample
Project\Models"
```

The user.ini file can be found in the following location: \MVRsimulation\ArcGIS Terrain Tools\Config\TerrainTools<*machinename*>user.ini.

Storing custom models for use in VRSG

If VRSG is installed on a terrain compilation machine, you can copy the given model to the VRSG installation folder where the other models reside, for easy access dueing a VRSG rendering session.

MVRsimulation recommends you store your models separately from the models installed with VRSG by creating a subdirectory for storing them, called \User, to keep them separate from VRSG's native models. For example:

```
\MVRsimulation\VRSG\Models\User
```

Be sure to edit the VRSGTerrainSearchPath.txt within the Models directory to add the \User subdirectory to the search path file so that VRSG will look in the directory.

Adding water surface areas

Using overhead imagery to represent water never results in realistic water regions in 3D terrain because water is highly reflective and often looks inconsistent in such imagery. Instead of using the overhead imagery to represent the water regions in your 3D terrain, you can add a water polygon feature to the terrain that will mark the areas covered by water.

Terrain Tools contains two built-in water feature definitions with different coastline treatments:

- MVRsimulation_Geometric_Water feature defines the coastline as cut out of the terrain geometry. This feature provides the best result for ground-level littoral/amphibious simulations and is recommended for most use cases. The terrain will render more efficiently at elevation because the feature geometry will be decimated for coarser LODs. VRSG will render areas assigned to this feature as 3D oceans, which means it will simulate waves, wakes, and other 3D ocean features.
- The MVRsimulation_Legacy_Water feature also defines the coastline as cut out of the terrain geometry. However, VRSG will render areas assigned with this feature with the older 2D normal mapped water texture. This is used in cases such as rivers, lakes, and other inland bodies of water where the elevation of the water surface differs from the global VRSG 3D ocean elevation. A future VRSG release will support variable water elevation with 3D oceans.

When the 3D terrain is built with the water feature, the appropriate areas will be marked as water surfaces in the resulting terrain tiles. When the resulting 3D terrain tiles are visualized in VRSG, these areas of water will be rendered with techniques that result in much greater visual fidelity. For example, VRSG will render sea-level water as 3D oceans with 3D waves and wakes, which is particularly useful for sea-based training scenarios. The default elevation for a 2D water feature is 0 MSL; by making the water a 3D feature, or by adding an "Elevation" attribute, you can assign it an elevation.

When you build the 3D terrain, the appropriate areas marked as water will be marked as a water surface in the resulting terrain tiles. When the resulting terrain is visualized in VRSG with 3D ocean simulation turned on, VRSG will render these sea-level water areas as a 3D ocean.

To try out the water surface feature see the chapter "Building 3D Terrain Tutorial."

Creating z-enabled (3D) polygons for water features: lakes, rivers, and floods

Water features such as lakes, rivers, and other standalone bodies of non-ocean water require the MVRsimulation Legacy Water feature definition and z-enabled polygon features that are assigned a uniform z-value defining the elevation of the water feature at which the water body resides. You can also simulate a flood by using a z-enabled polygon of a flooded area outlined on the terrain.

The MVRsimulation Legacy Water feature definition is required for these types of water features as Geometric Water 3D ocean water is not currently supported for these types of features. (Geometric Water is only supported at sea level.)

Water features that are assigned a Legacy Water feature definition are cut out of the terrain geometry and a 2D normal mapped water texture is applied, as shown below:

VRSG screenshot demonstrating lakes and rivers. These water features were compiled into the terrain using z-enabled polygons with z-values and the MVRsimulation Legacy Water feature.

Among the methods that exist for converting an existing 2D shoreline water feature to a 3D (z-enabled) polygon, one option is to use the 'Feature Class to Feature Class' tool found in ArcMap's Conversion Tools toolbox. On the Environments tab within the tool, you can change the output feature class to enable z-values on export. This will create a z-enabled polygon with z-values at 0. You will then need to add z-values to these polygons.

To add z-values to a z-enabled polygon using a raster surface as input, you can use the Update Feature Z tool (located in the 3D Analyst toolbox). This tool will automatically update z-information from an elevation raster surface to a z-enable polygon. To view and change a feature's z-values to create a uniform elevation of a lake, use the Edit Sketch Properties window (in the Editor toolbar) to edit the vertices. The z-values are listed in the Z column. To change the z-values for multiple vertices, check the boxes to the left of the vertices you want to update, then click the Z button and type the new values. If you want a completely flat surface, set the z-values to the same elevation value.

Creating light lobes and powerlines for streetlight or utility pole models

Populating a scene with large amounts of streetlight and utility features can sometimes lead to performance issues when the resulting 3D terrain is rendered in VRSG. To solve this problem, VRSG uses geometry instancing to render these features in VRSG more efficiently. The easiest and most scalable approach is to let Terrain Tools compile this instancing data directly into the mds tile itself. Alternatively, the user can use the Create Instanced Cultural Features tool to generate a .hpx/y model containing the instancing data. For the latter approach, the resulting models are geolocated in a cultural feature (vrsg.clt) file.

Instanced powerlines and utility poles on MVRsimulation's Hajin, Syria, 3D terrain.

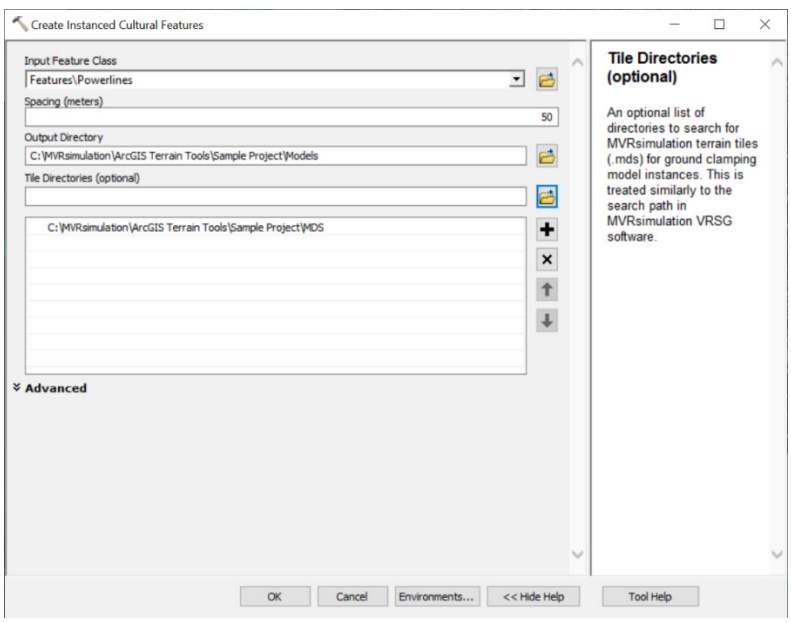
Powerline models with connecting wire geometry are created in Terrain Tools using a polyline feature. There are two different methods for generating powerlines in Terrain Tools as described below:

Polylines with vertex locations for pole placement:

When digitizing a polyline, each vertex location can indicate a pole placement. This workflow helps create powerlines in geospecific locations using imagery as a reference. After you have digitized your polyline, simply add a 'Model' field to its attribute table and assign a utility pole model with wire connectivity. (The models from MVRsimulation's 3D content libraries containing wire connectivity are listed below.) Place this feature under the Features group layer in the TOC. Terrain Tools will compile the poles with the wired connectivity directly in the terrain tiles.

Polylines with a specified distance for pole placement

Another option for creating powerlines is to specify a distance (in meters) for the pole placement. This method is useful when you have an existing polyline feature class with unknown pole locations or when you are unable to digitize the poles from the underlying imagery data. Add a 'Model' field to its attribute table and assign a utility pole model with wire connectivity. Use the Create Instanced Cultural Features tool located in the Features toolbox to generate a powerline model with a specified pole distance.



The output of this tool exports an instanced powerline model and a CLT file.

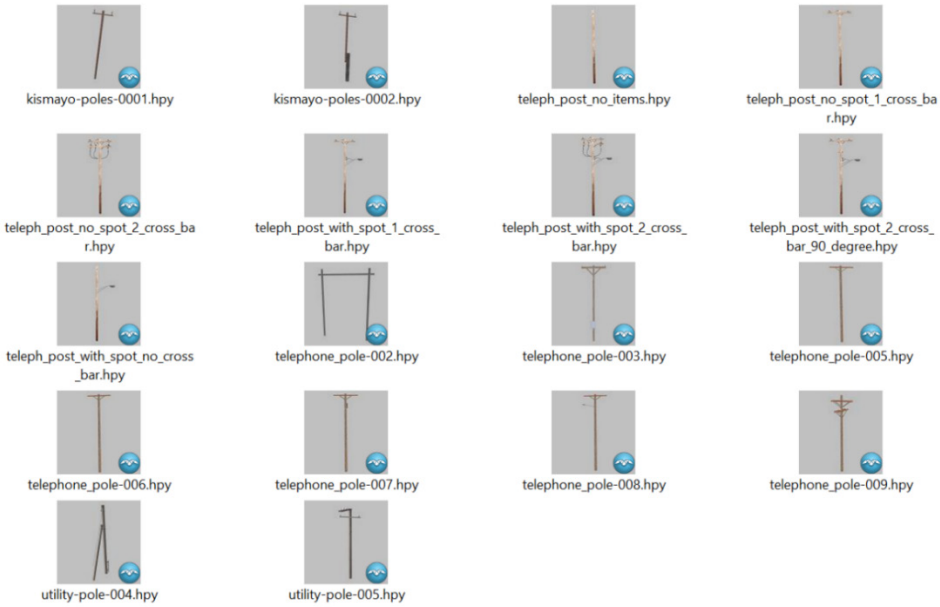
The tool will ground-clamp the model instances according to the elevation in the terrain tile(s) on which the utility poles will sit. You can choose to load the instanced hpx model with the vrsg.clt file in VRSG or you can import the CLT file using the Import Features From CLT File tool in the Utilities toolbox to create a point feature layer which you can then place under the Features group layer in the TOC. The instanced model will then be compiled into the terrain tile.

An example file is located in \MVRsimulation\ArcGIS Terrain Tools\Sample Project\Vector Data\Powerlines\Example\Powerlines.shp.

In MVRsimulation's 3D content libraries, the following models contain the wire geometry:

streetlight-003.hpy
streetlight-004.hpy
streetlight-015.hpy
streetlight-016.hpy
teleph_post_no_items.hpy
teleph_post_no_spot_1_cross_bar.hpy
teleph_post_no_spot_2_cross_bar
teleph_post_with_spot_1_cross_bar.hpy
teleph_post_with_spot_2_cross_bar.hpy
teleph_post_with_spot_2_cross_bar_90_degree.hpy
teleph_post_with_spot_no_cross_bar.hpy
telephone_pole-002.hpy
telephone_pole-003.hpy
telephone_pole-005.hpy
telephone_pole-006.hpy
telephone_pole-007.hpy
telephone_pole-008.hpy
telephone_pole-009.hpy
utility-pole-004.hpy
utility-pole-005.hpy

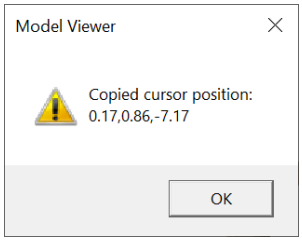
Also, kismayo-poles-0001.hpy and kismayo-poles-0002.hpy, included with the Kismayo, Somalia 3D terrain.



Adding wires to custom models using a JSON file

You may want to add wires to your own custom models or to pole models in the MVRsimulation model libraries ibraries that do not already contain them. To do so, you can create a JSON file using Model Viewer to define the attachment points for the wire placement.

Wires can be added to a model by using Model Viewer to define the wired connectivity. When you open a model in Model Viewer, position the mouse cursor over the desired wire connection, and then press the P key on the keyboard. This automatically copies the wire location to your Windows Clipboard. The following confirmationmessage appears with the x,y,z locations:



Next, open a text document in Notepad (or another text editor) and paste the Clipboard data. If there is another wire connection, place the cursor over the desired area and again press the P key and paste the data into the text editor.

The example below contains two copied cursor positions pasted into a text editor:

0.17,0.86,-7.17

-0.00,1.13,-11.83

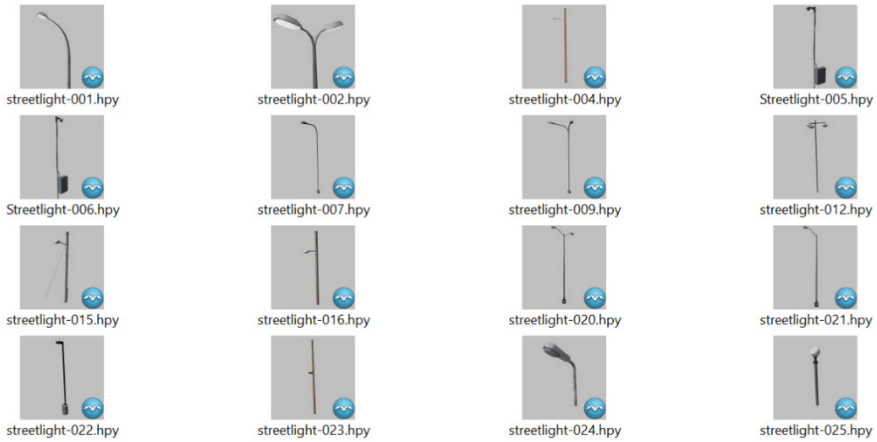
Next, you need to add the formatting for a wired connection. The formatting is outlined below with the values above copied into the x,y,z locations:

```
{
  "version": 1,
  "wire": [
    {
      "offsetX":   0.17,
      "offsetY":   0.86,
      "offsetZ":   -7.17
    },
    {
      "offsetX":   0.0,
      "offsetY":   1.13,
      "offsetZ":   -11.83
    }
  ]
}
```

Save the text document with a .json file extension and copy the .json file to the directory where your HPX (or HPY) model is located. Terrain Tools is now ready to compile the instanced wired pole geometry into an HPX model. Run the Create Instanced Cultural Features Tool as described in the workflow above.

Streetlight models with light lobes

When a light lobe is attached to a model, VRSG will render the lobe during a night scene in the VRSG IG. The streetlight models that are shown in the graphic below also have built-in light lobe attributes compiled into the HPY model.

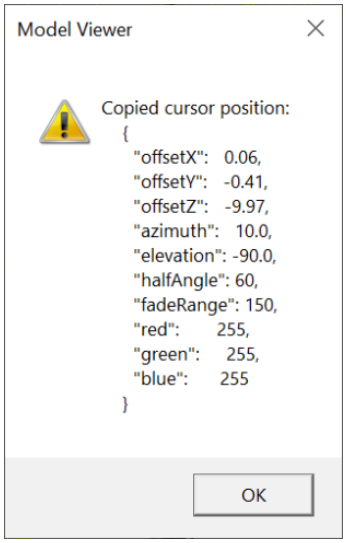


You can drag and drop these models on the terrain in a VRSG window to preview their light lobe. On the Environment tab on the VRSG dashboard, change the Time of Day to dusk. The streetlight models are located in \MVRsimulation\VRSG\Models\Other.

Other models from our MVRsimulation Model Library containing light lobes include stadium lights and various airport models.

Adding light lobes to custom models using a JSON file

You may want to add light lobes to your own custom models or to models in the MVRsimulation Libraries that do not already contain light lobes. To do so, you can create a JSON file using Model Viewer to define the attachment points for the light lobe placement.Light lobes can be added to a model by using Model Viewer. When you open a model in Model Viewer, position the mouse cursor over a light fixture, and then press Shift P. This action copies the light lobe JSON information to your Window's Clipboard and displays the following message:



Next, open a text document in Notepad (or another text editor) and paste the Clipboard data. Edit the copied light lobe information within the JSON code as shown below:

You can change the color of the light lobe by changing the RGB values.

RGB(255,255,255) indicates a white lobe.

```
1  {
2    "version": 1,
3    "lightLobe": [
4      {
5        "offsetX":   1.05,
6        "offsetY":   -0.01,
7        "offsetZ":   -4.28,
8        "azimuth":   10.0,
9        "elevation": -90.0,
10       "halfAngle": 60,
11       "fadeRange": 150,
12       "red":       255,
13       "green":     255,
14       "blue":      255
15     }
16   ]
17 }
18
```

If you have more than one light lobe per model, you would need to separate the lobes by using a comma and make sure that the JSON formatting is correct. An example is shown below:

```
1  {
2    "version": 1,
3    "lightLobe": [
4      {
5        "offsetX": 0.02,
6        "offsetY": -0.77,
7        "offsetZ": -1.74,
8        "azimuth": 10.0,
9        "elevation": -90.0,
10       "halfAngle": 60,
11       "fadeRange": 150,
12       "red": 255,
13       "green": 0,
14       "blue": 0
15     },
16         {
17       "offsetX":   -0.01,
18       "offsetY":   0.71,
19       "offsetZ":   -1.73,
20       "azimuth":   10.0,
21       "elevation": -90.0,
22       "halfAngle": 60,
23       "fadeRange": 150,
24       "red":       255,
25       "green":     0,
26       "blue":      0
27     }
28   ]
29 }
```

After you have completed editing the file, save it using the model's name with a .json extension. Next, copy the .json file to the directory where your hpx (or hpy) model is located. To preview the light lobe(s), you would just drag the hpx that is in the same directory as the .json file into VRSG and change the time of day to the darkest setting. (Light lobes do not render in Model Viewer.) Terrain Tools will compile the instanced light lobe(s) into the terrain when the feature is placed in the Features group layer.

Elevation features

There are several ways to fine tune the elevation in your 3D terrain project. You can choose to modify the source elevation using one of several tools provided by Esri in the ArcToolbox or you can simply modify the terrain through the use of elevation features. Elevation features are the simplest way to modify or smooth underlying elevation data.

Two types of elevation features can be used for modifying underlying source elevation: polygon and point features.

Polygon elevation features

Polygon features will mask out terrain vertices from the interior. This workflow is useful for flattening or smoothing elevation data underneath a runway model.

- Using a z-enabled polygon (3D), the z values of the boundary will sample from the z-values that are assigned in the feature.
- Using a non-z-enabled polygon (2D), the z values of the boundary will sample from the underlying elevation source data.

Point or multipoint elevation features

Point features can also be placed within a polygon feature or any location on the terrain to fine-tune any area of the terrain.When a z-enabled point feature is used, z-values defined in the feature are used to further refine elevation in a specific area. Like polygon features, multipoint features will mask out terrain vertices inside the multipoint bounding polygon.

Editing features that have z-values

Each vertex of a feature can store a z-value along with its x,y positional information. To store z-values in a feature, you must specify that the feature class has z-values when you create it. This makes the feature class z-aware. To determine if a feature class is z-aware, open its attribute table. In the Shape field of the attribute table, a Z will appear in the field.

To add z-coordinates to your z-enabled polygon vertices using a raster surface, you can use the Update Feature Z tool (3D Analyst). This will automatically update z-information from an elevation raster surface to a z-enable polygon.

You can also view or change a feature's z-values by using the Edit Sketch Properties window when editing vertices. The z-values are listed in the Z column. To change the z-values for multiple vertices, check the boxes to the left of the vertices you want to update, then click the Z button and type the new values. If you want a completely flat surface, set the z-values to the same elevation value.

Elevation feature placement in the TOC

Elevation features can be placed in the Features group layer or the Elevation group layer in the TOC depending on your use case. By placing these features in either group, Terrain Tools will interpret them differently.

- Features group layer - When elevation features are placed under the Features layer, they are cut into the terrain skin. Vertices are introduced to the terrain skin for each vertex in the feature. The elevation of the enclosed area will be interpolated area based on the Z value for each vertex.
- Elevation group layer - If an elevation polygon is placed in the Elevation group layer in the TOC, the feature will not be cut into the terrain. Instead, it will be treated similarly as a raster elevation source. Elevation data will be sampled from the feature, but no new vertices will be introduced. Note that the layer order still takes precedence in this case.

Bathymetry

Bathymetry data is the elevation data of the ocean floor. Compiling bathymetry data in the terrain improves the fidelity of the 3D water in the shoreline area. Particularly, waves will dampen in shallower water, and water will become more transparent. This is useful in cases where VRSG is used for ground-level littoral simulations.

By default, Terrain Tools will automatically compile a geotypical procedural bathymetric shoreline effect into water features when assigned an MVRsimulation_Geometric_Water feature definition and when placed under the Features group layer in the TOC. To use the default procedural bathymetry depth, nothing further is necessary.

When using procedural bathymetry produced by using a Geometric water feature, the bathymetry terrain surface depth is computed for each terrain vertex as a function of the distance to the closest point on the coastline.

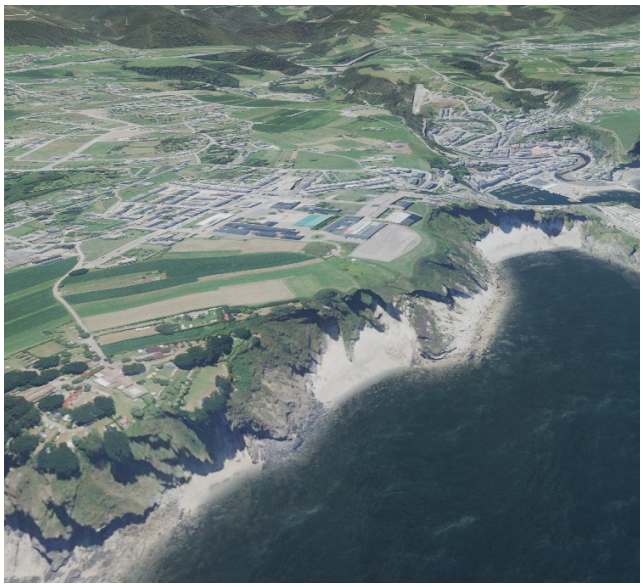
However, you might want to compile geospecific bathymetric source data into your project. For this option, you can add bathymetric source data to the Bathymetric group layer in the TOC. The resulting underwater terrain surface is essentially identical to terrain but with a geotypical sand microtexture applied instead of imagery.

If needed, it is also possible to fine-tune the bathymetry elevation and vertices using similar methods to regular terrain. However, when you place such features in the Features layer, there is an additional step you must take to identify the feature as operating on the bathymetry surface. For any polyline/point/polygon terrain modifying feature, you will need to add a FeatureDef field and the attribute value should be set to "BathyTinFeature".

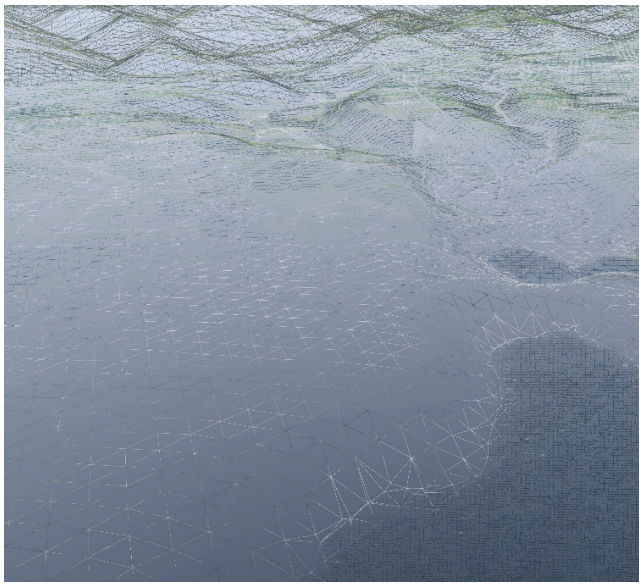
The Bathymetry feature layer is available in the TOC as part of the MVRsimulation map template. However, to add supplemental or geospecific bathymetry data to a project:

1. In the Catalog panel, locate the bathymetry source raster you want to use.
2. Select and drag the bathymetry raster to the TOC and place it in the Bathymetry layer.
3. Specify the MVRsimulation_Geometric_Water feature type for polygonal water features that interact with the bathymetry source data. (Shoreline vectors used in conjunction with bathymetric data must be 3D (z-enabled) polygons; otherwise, a gap will appear between the shoreline and the ocean.)

During terrain tile generation, Terrain Tools assigns a texture called BeachSand-03.tex to the shoreline. It will also create a \Textures subdirectory within the terrain tiles output directory and copy BeachSand-03.tex to that directory. During visualization, VRSG will automatically search the \Textures directory for the BeachSand-03.tex texture file when bathymetry data is compiled into a terrain tile.



Procedural bathymetric shoreline example of Costa Verde, Spain, rendered in VRSG, showing the ocean wave fall-off and transparency at the shoreline.



VRSG's wireframe view of the same scene on the left, in which you can see the shoreline and underwater modeling.

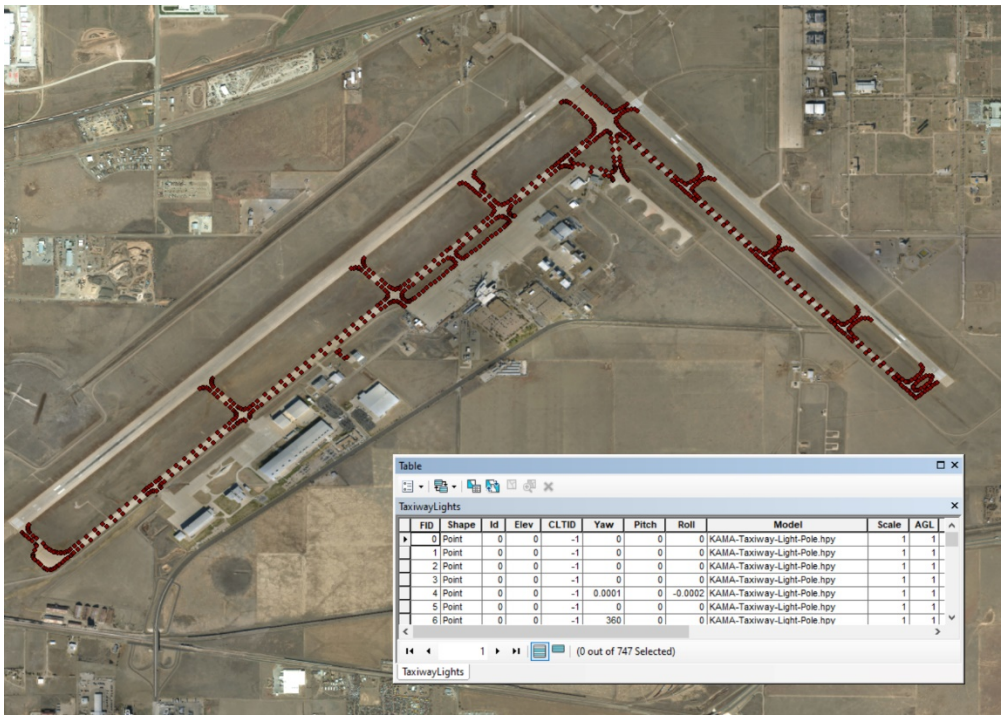
Shoreline polygon features can be 2D or 3D. For 2D features without an Elevation attribute, the elevation defaults to 0 MSL.

Importing features from a cultural feature file

You can import model locations from a cultural feature file (vrsg.clt) to a point feature class layer by using the tool Import Features From CLT File. This tool is located in the Utilities toolset.

A vrsg.clt file is a cultural feature file that specifies a model's location and some of its appearance attributes such as scale on the terrain. This file can be loaded in VRSG, Scenario Editor, or Battlespace Simulation's (BSI's) Modern Air Combat Environment (MACE) application. A vrsg.clt file can be generated by placing models on the terrain using any of these programs. Each application enables you to export vrsg.clt files which you can then import into Terrain Tools for further analysis or editing. You can also create a vrsg.clt file by hand in Notepad with required formatting, which is explained in the *MVRsimulation VRSG User's Guide*.

Importing features from a cultural feature file in Terrain Tools enables you to visualize the geolocation of each culture feature as a point feature on the imagery in your map document. The point locations represent the origin of each 3D model. You can edit the placement of points and/or modify their attributes such as scale or yaw in an active edit session.



Taxiway model locations are represented as point features in the map document after importing them from a CLT file. The attribute table shows the CLT attributes.

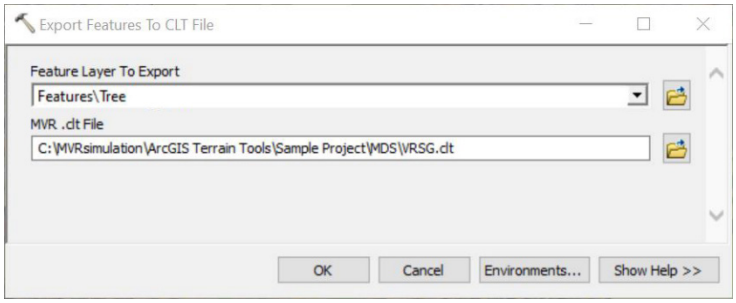
The most commonly used components of a CLT file are described briefly below and in full in the *MVRsimulation VRSG User's Guide*.

- Model − Text attribute that specifies the name of the model (in MVRsimulation's HPY or HPX format) that will be used to visualize the point feature in VRSG.
- Elev − Floating point attribute that specifies the elevation (z-value). The interpretation of this value depends on the AGL attribute.
    - If there is an AGL attribute field the is set to 1 (meaning AGL is on), the Elev value is used as above ground level for the given point feature. Therefore, an Elev value of '0' combined with an AGL equal to1 will automatically clamp the model to the elevation of the terrain at the origin of the model.
    - If there is an AGL attribute field the is set to 0 (meaning AGL is off), the Elev value is used as MSL (meters above sea level). Therefore, an Elev value of '1250' combined with an AGL equal to 0 will load the model at its origin at the elevation of 1250 meters.
- Yaw, Pitch, and Roll − Floating point attributes specifying the orientation of the point feature in terms of rotations around the Z, X, and Y axes respectively.

- Scale − Floating point value that specifies a uniform scaling factor that will be applied to the model that represents this point feature. A scale value of 1 indicates the model's original size.
- AGL – Integer attribute for determining how the Elev field is interpreted. A value of 1 means the Elev value is used as meters above the underlying terrain. This is also the default if no AGL field is defined. A value of 0 means the Elev value is used as MSL.
- CLTID – Integer value to define the entity ID for the entity. If there is no CLTID field, the IDs will be generated automatically. If -1 is provided as a value, no CLTID will be output (for static entities).
- LOD_Scale – Floating point value is used by VRSG to scale the LOD ranges of the model. A LOD scale of 1 indicates the model's original LOD scale.

Exporting features to a cultural feature file

You can also export an MVRsimulation cultural feature file (vrsg.clt) that contains model locations from a given point feature layer by using the tool Export Features to CLT File. This tool exports the features from a given feature layer to a new MVRsimulation cultural feature file (vrsg.clt).To export features to an MVRsimulation cultural feature file, in the Utilities toolset (located in the MVRsimulation Tools toolbox), double-click the Export Features To CLT File tool.



The exported cultural feature file can be used as input to:

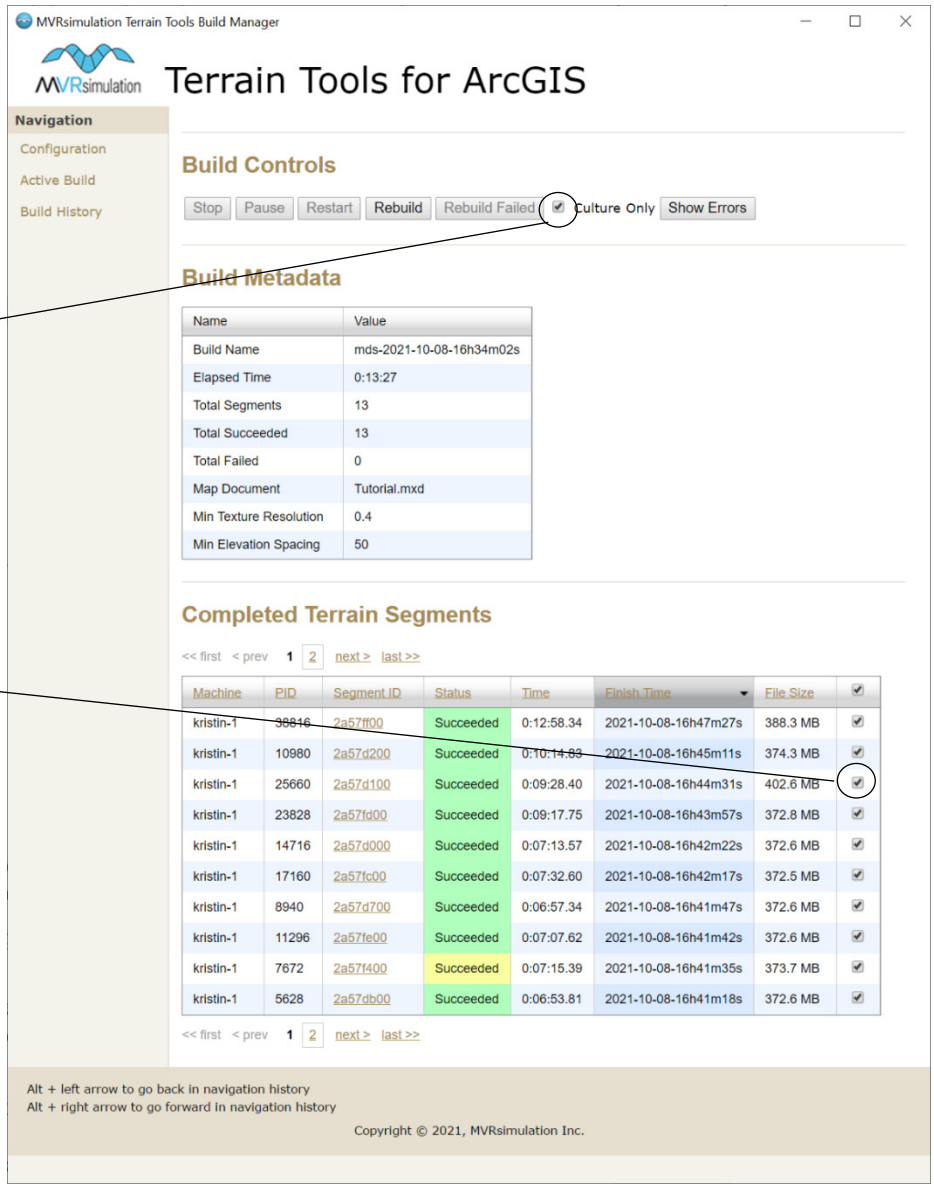
- Load a vrsg.clt file into VRSG to your render models on the terrain.
- Import a vrsg.clt file in VRSG Scenario Editor to edit models interactively on the terrain. Scenario Editor can then export your modifications to a new CLT file.
- Import a .clt file into Battlespace Simulations' third-party software MACE for use in a CGF/SAF environment.

Rebuild culture without recompiling tile geometry

You can recompile some cultural features of one or more 3D terrain tile(s) without rebuilding the entire terrain geometry and textures. This feature is useful for when you are iterating on cultural feature work for a given 3D terrain and would like to recompile the changes quickly, bypassing the time it takes to process terrain geometry for the tiles on which the cultural features are located.

Rebuilding the cultural features on a tile selection will rebuild features that sit on the terrain (buildings, trees, fences, powerlines, and cultural lights). It will not rebuild features that cut into the terrain (such as roads, runway models, or bodies of water).

In the Build Manager history, you can browse to a build in the Build Manager, select all or specific tiles, and click the Rebuild button.



Select this checkbox to only rebuild the cultural features on an existing tile or set of tiles. (This option does not work for cut-in features such as water, roads, or runway models.)

Select or unselect tiles to be rebuilt. The checkbox at the top will select all the listed tiles.

Battlespace Simulations' Modern Combat Environment (MACE) and terrain feature data

If you build terrain that will be used in a simulation system with Battlespace Simulations' Modern Combat Environment (MACE) with VRSG, bear in mind that MACE can import the source vector layers that represent roads, buildings, and water locations on a 2D map in MACE. You can import any vector files into the MACE that were used as source data in Terrain Tools so that the correlated terrain features will be shown in MACE.

- With road vectors, MACE can have vehicles follow the roads.
- With building vector, MACE can have characters automatically walk around buildings.
- With water vectors, MACE can change the detonation PDU of munition to have water splash effects rather than dust.

Cultural feature files you export from Terrain Tools can also be imported into MACE.

Working with OpenStreetMap data

One popular source for vector data for terrain compilation projects is OSM data. The Terrain Tools tutorial uses OSM data for the Baltimore building footprints and road and light point vectors. The most common format for OSM distribution is the XML-based .osm format. ArcGIS does not natively load this format. However, there are several ways to convert this data into Esri's geodatabase or shapefile formats.

For small areas of data, the easiest approach is to use one of the many online distributors that will extract a subset from the OSM database and convert it to a file format compatible with ArcGIS. The disadvantage is OSM imposes a limit on the size of the download when extracting data. This might not be sufficient when dealing with a large or very dense area of data.

Alternatively, it is possible to download a compressed binary copy of the entire OSM database, called a *planet* file from https://planet.openstreetmap.org/. This data is distributed in the .pbf format and is much smaller and faster to process than the XML .osm format. Subsets of this data can be extracted, filtered, and converted back into the .osm format using one or more open-source command-line tools maintained by OpenStreetMap. The tools OsmConvert, Osmosis, and OsmFilter are useful for working with the data. See http://wiki.openstreetmap.org for information about these tools.

When you work with OSM formats directly, it is important to keep in mind that due to its generality, the data can be overly broad and bulky for your use case. For instance, if you are only interested in building footprints, you should filter out all other data (such as roads, land use, or water) early on in your workflow. Doing so can have a large impact on processing time.

To convert the data from .osm format to an ArcGIS feature class, there are multiple options as well. Esri maintains an open-source extension called OSM Editor for ArcGIS, which you can obtain at http://www.esri.com/software/arcgis/extensions/openstreetmap. This extension can be used to extract OSM ways and nodes into points, polylines, and polygon feature classes. The extension can also create symbolized layer files, which is useful for making a distinction between many types of polygon and polyline data.

Implying terrain feature attributes using a joined table

For large feature datasets, it is sometimes easier to use an existing attribute to imply the feature definition to avoid modifying the data. You can accomplish this by doing a join with a standalone table. For example, OpenStreetMap (OSM) data has a wide range of attributes describing the data, such as the Highway attribute for roads, and the Building attribute for buildings. You can use these attributes to join a standalone table containing the FeatureDef names, as described in the following steps:

1. Create a new table and name it "FeatureDefMapping."
2. Use the Data Management >Fields >Add Field tool to add a text field that will contain the values of the existing feature class attribute. You can name the field anything, but a name like "osm_key" would be descriptive.
3. Run the Add Terrain Attributes tool using this table as input to add the FeatureDef attribute to this table.
4. In ArcMap, right-click on the table, and choose Edit Features >Start Editing. Open the attribute table and define the desired mappings. For example, if the vector data had an existing attribute Highway describing the road type, and a value "secondary", you would add a row to the FeatureDefMapping table with the osm_key attribute set to "secondary" and the FeatureDef attribute with the desired FeatureDef name.
5. In ArcMap, add the existing vector data to the map document. Right-click on it, and select Joins and Relates > Join….
   a. Select the attribute used to imply the feature definition in the first box.
   b. Browse to or select the FeatureDefMapping table in the second box.
   c. Choose the osm_key field to base the join on, and click OK.

Now the existing data will render a FeatureDef attribute based on the join. Note that this is stored in the map document, not in the vector data itself.

CHAPTER 5

Managing Terrain Builds

Distributing 3D terrain generation work across multiple machines is an effective way to increase terrain yields. The Terrain Tools Build Manager automates much of this process, eliminating manual steps and providing a user interface.

Note: Although the minimum RAM requirement for running Terrain Tools is Esri's minimum requirement of 8 GB, MVRsimulation recommends 16 GB or more RAM for compiling terrain. The higher the resolution of source data, both raster and vector data, the more RAM is required for compiling the terrain.

Overview

The Build Manager is a standalone application that runs in the background on each build machine and has two categories of responsibilities, both of which are optional:

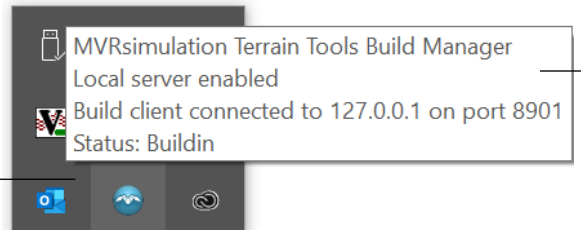
- Build server − manages terrain-building requests, and centralizes the build data (that is, the set of tiles to build, as well as any build parameters). Build clients from all participating machines register with this process. The build server also serves data about the build for consumption by the Build Manager UI process.
- Build client − manages asynchronous build processes for this machine.

In a simple single-machine build environment, the Build Manager acts as both server and client. Additional client machines connecting to the server machine only need to use the build client functionality of Build Manager.

The Build Manager interface is a web browser application in which you can configure and manage both the build server and the clients(s), as well as any actively building database.

Note: Internet connectivity is not necessary to build terrain.

When the Build Manager is running, its icon is displayed in the Windows system tray:



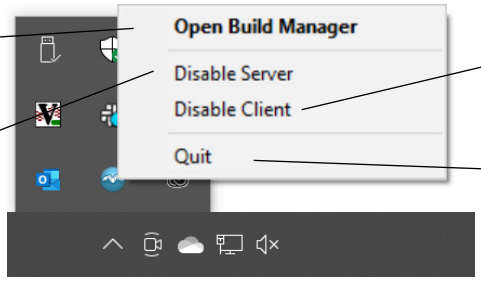
Information that appears when you hold the cursor over the Terrain Tools Build Manager icon.

Terrain Tools Build Manager.

You open the Build Manager by double-clicking the icon in the system tray, clicking Terrain Tools Build Manager in the Start Menu, or by right-clicking the icon and choosing Open Build Manager from the menu. Right-clicking the icon opens the following menu:

Opens the Build Manager window for configuring the build server and client(s) and administering the active build.

Toggles enabling/ disabling the server functionality for this machine.



Toggles enabling/ disabling the client functionality for this machine.

Ends the Build Manager process and stops any active terrain-building process.

Double-clicking on the icon also opens the Build Manager window.

Using the Build Manager

The Build Manager window is where you manage the build configuration and monitor build status. You can administer all connected terrain-building machines from this centralized location.

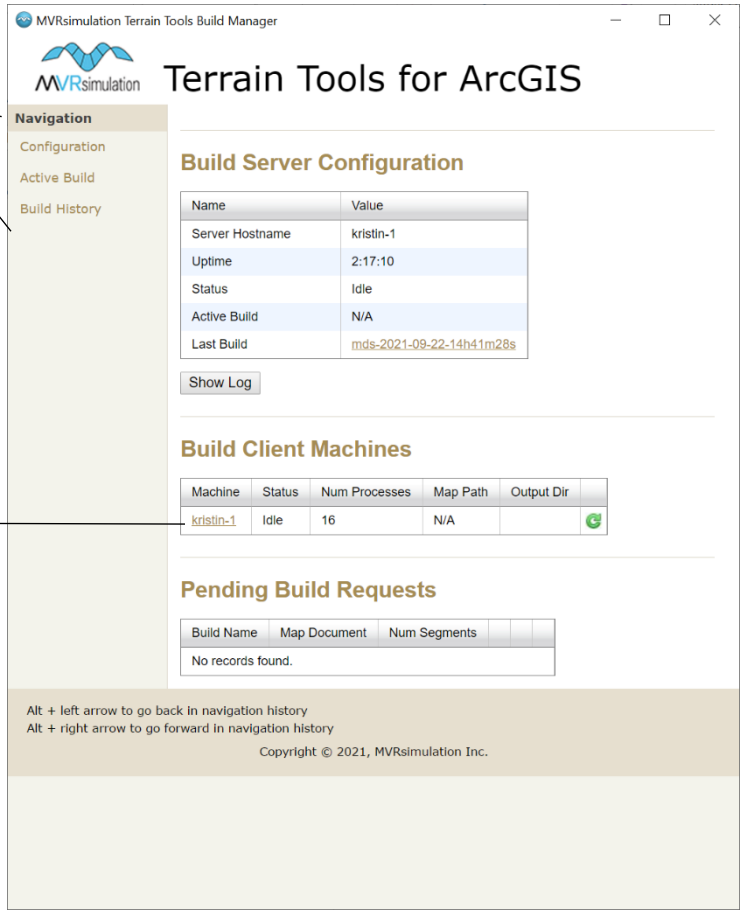
If the Build Manager is not running as a background process when you start a terrain building process (for example from using the Build Selected MDS Tiles or Create Selected MDS Tiles Build Script tool, you might need to start it manually. To do so, locate "Terrain Tools Build Manager" in the Windows Start menu and double-click it. The first time you use Terrain Tools is one case in which you would need to start the Build Manager manually.

Server configuration

When the Build Manager is first opened from the Windows system tray, it loads the configuration view, as shown in the following example. The Build Manager window has navigation links on the left which lead to three primary views: Configuration, Active Build and Build History. The main section of the window contains content pertaining to the given topic.

The main body of the Configuration page contains the following three sections:

- Build Server Configuration contains a table of server metadata. The Show Log button displays a textbox containing logging information about the server.
- Build Client Machines contains a table listing all build machines connected to the server. The Machine column contains a hyperlink for each machine. Click on the link for a machine to open the Build Client Configuration page for that machine.
- Pending Build Requests lists the active request and any pending build requests received by the server. There can only be one active build request on the server.



Navigation panel.

List of client machines. Click to configure.

Client configuration and preferences

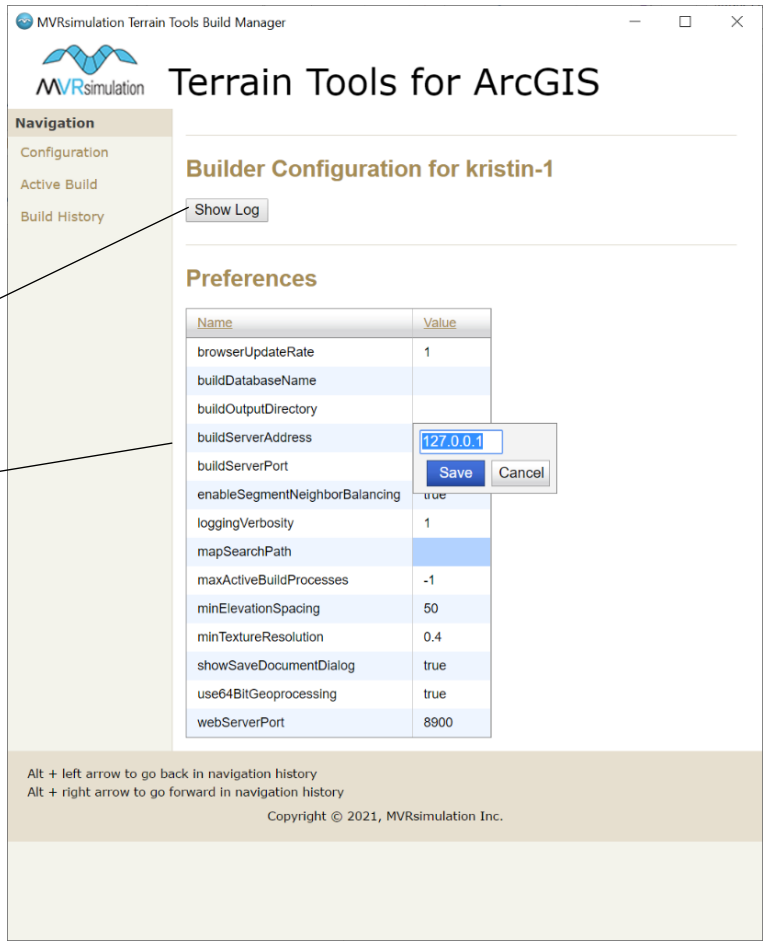
The client configuration page contains two sections:

- Builder Configuration − press the Show Log/Hide Log button to display/hide a text box containing logging information about the client.
- Preferences − a table containing a row for each preference entry in the Terrain Tools preferences file.

Terrain Tools preferences are stored in \MVRsimulation\ArcGIS Terrain Tools\Config\TerrainTools_default.ini. You can make changes to preferences in two ways:

In an ASCII text editor like Notepad, edit the file \MVRsimulation\ArcGIS Terrain Tools\Config\TerrainTools_*USERNAME*_user.ini. For a given preference, use the syntax exactly as used in the TerrainTools_default.ini. These changes you make in the TerrainTools_*USERNAME*_user.ini affect *all* terrain builds associated with this Terrain Tools license.

In the Build Manager, change the preference values on the Client Configuration for a given machine, as shown in the example below. These preference changes affect *only* the client machine for which you have changed the preference values.



Click to display o hide the terrain-building log for the given client.

Click to edit the value of one or more preferences.

To edit one or more Terrain Tools preferences for a given machine:

1. Click the Value column header. Doing so puts the Values column in edit mode.
2. Click on a given preference value field. This opens a dialog box with a field where you can enter the new preference value.
3. Enter the new value and then click Save. This edit updates the preference value in the user configuration file in the installation directory (TerrainTools_*USERNAME*_user.ini).

The following table describes Terrain Tools preferences available in the Build Manager. See the appendix "Terrain Tools Preferences" for a list of *all* Terrain Tools preferences.

Preference	**Value**
auxExternalTexturePath	A \Textures directory for VRSG to search for textures for compiled terrain in addition to the default \Textures directory of the Terrain Tools or VRSG installation.
browserUpdateRate	Elapsed time in seconds between calls to refresh the dynamic data in the Build Manager window. (Default is 1 second.)
buildOutputDirectory	Output directory where compiled terrain is stored.
buildServerAddress	Hostname or IP address of the build server. (Default is 127.0.0.1, or localhost.)
buildServerPort	Port the build server uses. (Defaults to 8900.)
cultureFadeRange	Range, in meters, of LOD switch-out-distance at which cultural lights, fences, buildings fade out on the terrain. Default is 15,000 meters.
enableSegmentNeighbor Balancing	Enables the terrain segment neighbor balancing algorithm. (Default is true.)
loggingVerbosity	Detail-level of logging information output for each tile. 0=disabled, 1=metadata only, 2=metadata, and performance stats, 3=metadata, stats, and verbose debugging output. (Default is 1.)
mapSearchPath	List of semicolon-delimited path entries, each of which is used to locate the ArcMap .mxd document for the active build.
maxActiveBuildProcesses	Limit on the number of simultaneously running build processes. (Default is -1, which lets the client determine the maximum number based on the number of processors on the machine.)
minElevationSpacing	Default minimum resolution of elevation used. *Note:* Only used to fill in defaults in the ArcToolbox tool parameters.
minTextureResolution	Default minimum resolution of imagery used. *Note:* Only used to fill in defaults in the ArcToolbox tool parameters.
modelSearchPath	List of semicolon-delimited path entries, each used to locate models for the active build which are not stored in MVRsimulation's model libraries (\MVRsimulation\VRSG\Models).

Preference	**Value *(continued)***
showSaveDocumentDialog	Enables the helper dialog box to remind users to save the map when initiating a build from within ArcMap. (Default is true.)
webServerPort	Port that the Build Manager window's data server runs on. (Default is 8901.)

Active Build page

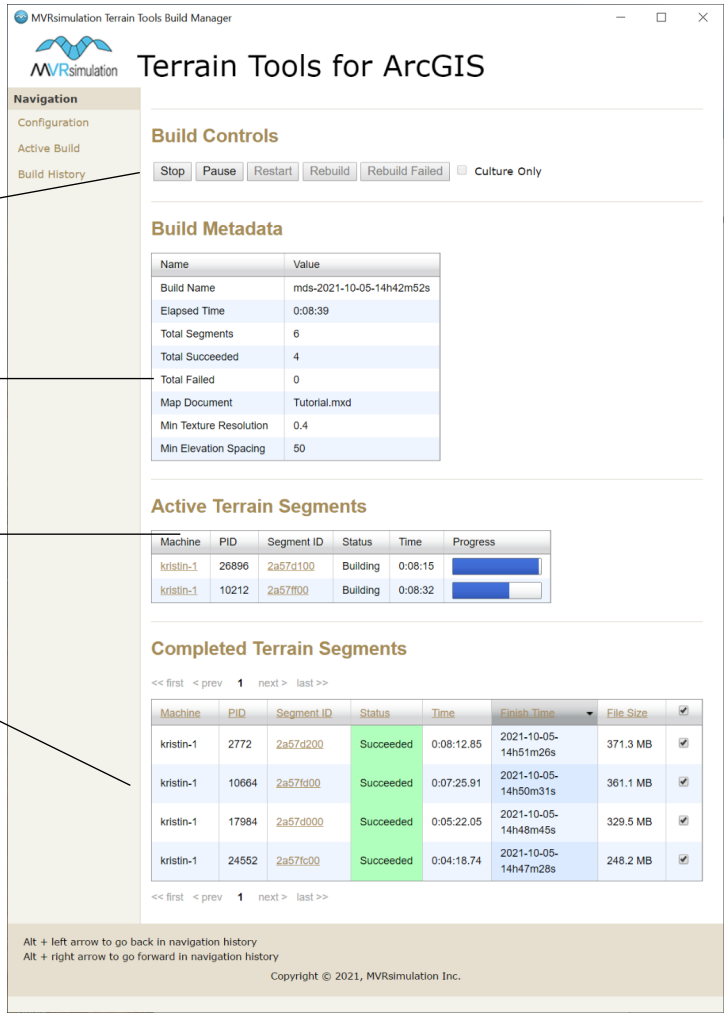
The Active Build page displays dynamically updated information about the current build running (or most recently completed) on the Build Manager.

Click one of these buttons to stop, pause, resume, or rebuild the active terrain-building process.

General information about the active terrain building process.

Information about all active terrain-building processes.

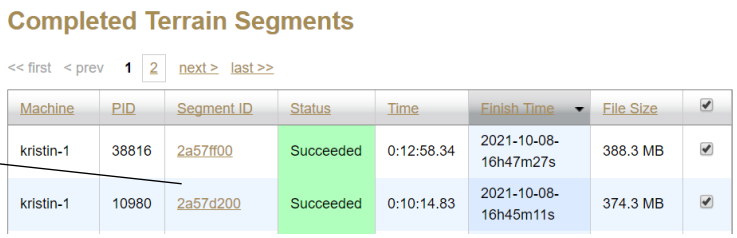
Information about all completed terrain builds.



For an active terrain build, the page displays three sets of information and a set of buttons for controlling the build.

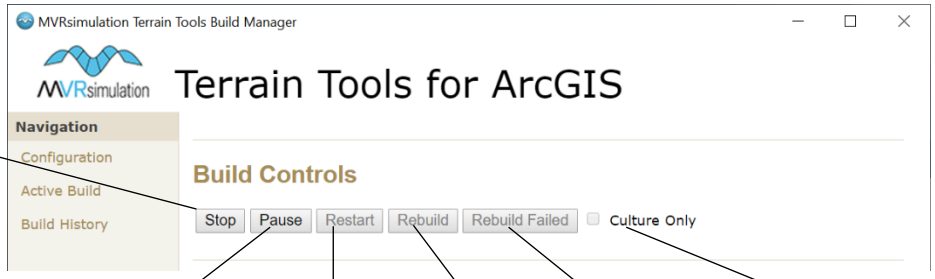
For each entry in the Active and Completed Terrain Segment tables, the Segment ID field contains a link to display the information log for building that terrain tile (segment).

Click to view the log for a particular terrain tile ("segment").



Use the buttons at the top of the page to control the build processes:

Stops the active build process. If there are any pending build requests, the oldest one will become the new active build.



Suspends the active build, while maintaining its status as the active build.

Generates a build request to resume a previously stopped build from where it left off.

Generates a build request to rebuild all the completed tiles.

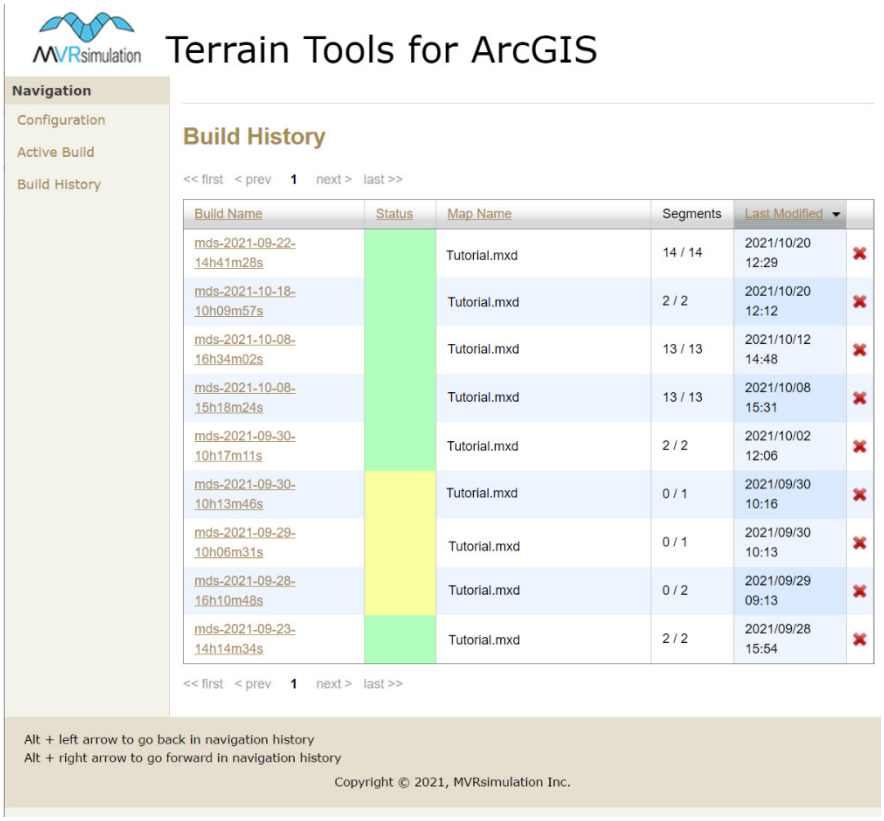
Generates a build request to rebuild only the failed tiles.

Generates a build request to rebuild all culture for completed tiles in a selected build in history.

You can rebuild just the cultural features that sit on top of the terrain tiles, without rebuilding the tiles themselves. While iterating on cultural feature work for a given set of tiles, this capability fulfills the need to recompile the changes quickly, bypassing the time it takes to rebuild the tiles on which the cultural features are located. Culture Only rebuilds features that sit on the terrain (buildings, trees, fences, and lights). It will not rebuild features that cut into the terrain (such as roads, runways, or bodies of water).

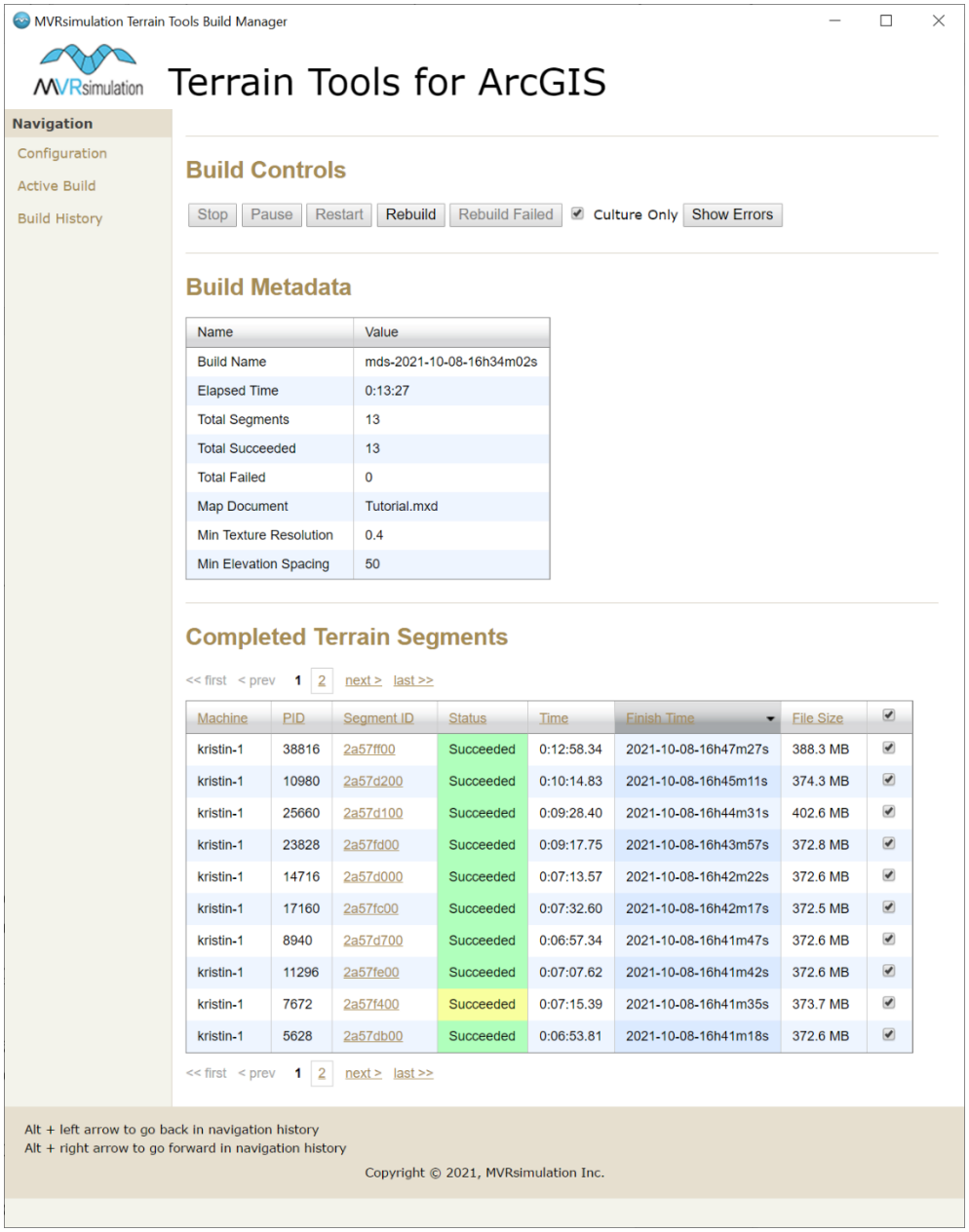
Build History page

The build data from each build request is stored on the build server on disk. The Build History page contains a table with an entry for each build request, referred to as a build name on the page. The build name includes a timestamp of when the build was submitted.



Click the red 'x' icon at the end of a row in the table to remove a particular build from the history. This action deletes all logging records for that build.

Click a build link in the Build Name column to display a page similar to the Active Build page (without the dynamic data), showing information about the terrain tiles ("segments") associated with that specific build request.



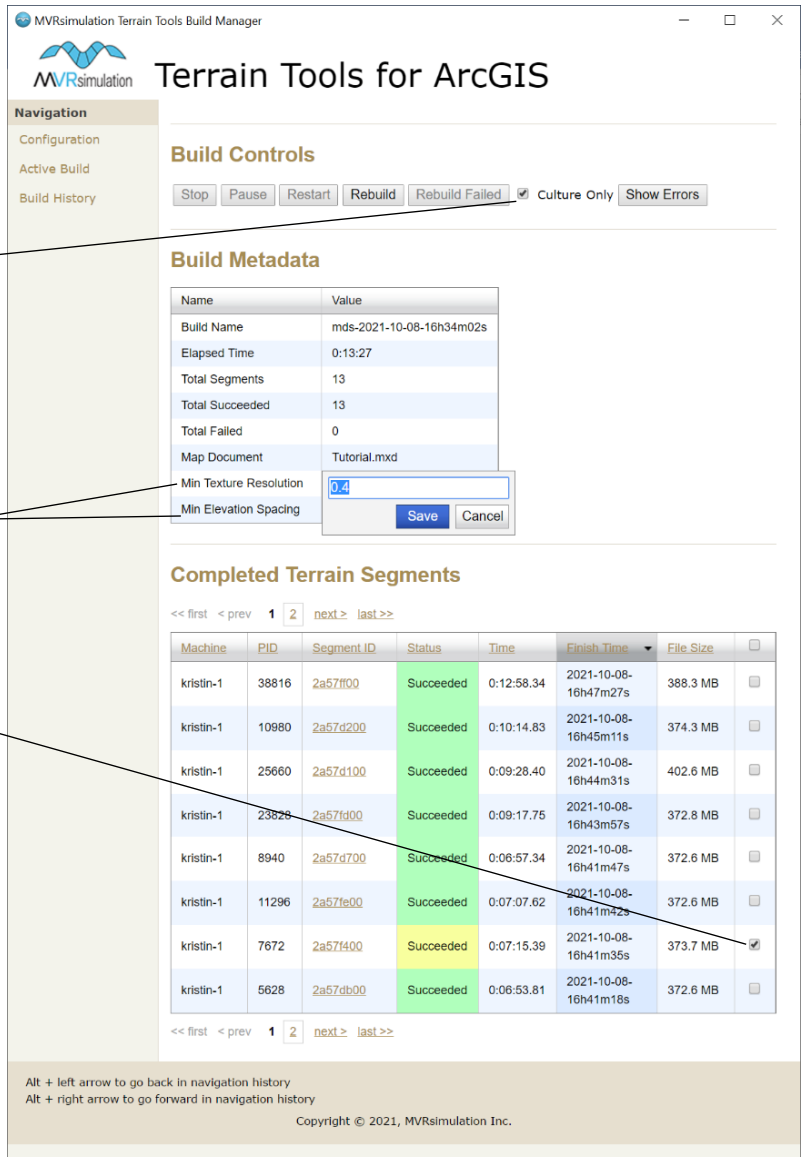
Rebuilding tiles

In the Build Controls section, click the Rebuild or Rebuild Failed buttons to issue a request to rebuild completed tiles or failed terrain tiles. Optionally, click the Culture Only checkbox to only rebuild the culture on the tiles, as described earlier.

New in Terrain Tools v2.0 is the ability to rebuild one or more selected tiles from an existing build, rather than rebuilding all the tiles in a build request. Optionally you can rebuild the selected tiles with different a minimum texture and/or elevation resolution.

To rebuild one or more selected tiles:

1. In the Completed Terrain Segments table, select the terrain tile or tiles to rebuild by clicking the checkbox at the end of the row of the tile(s).
2. Optionally do one of the following to specify changed details of the rebuild operation:
   - Click the Culture Only checkbox to rebuild only the culture for the selected tile(s).
   - In the Build Metadata table, click the Value field to change the minimum texture resolution and/or minimum elevation spacing for the selected tile(s) to rebuild.
3. In the Build Controls section, click the Rebuild or Rebuild Failed buttons to issue a request to rebuild one or more selected completed tiles or failed terrain tiles.



Click the Culture Only checkbox to rebuild only the culture on the selected tile(s).

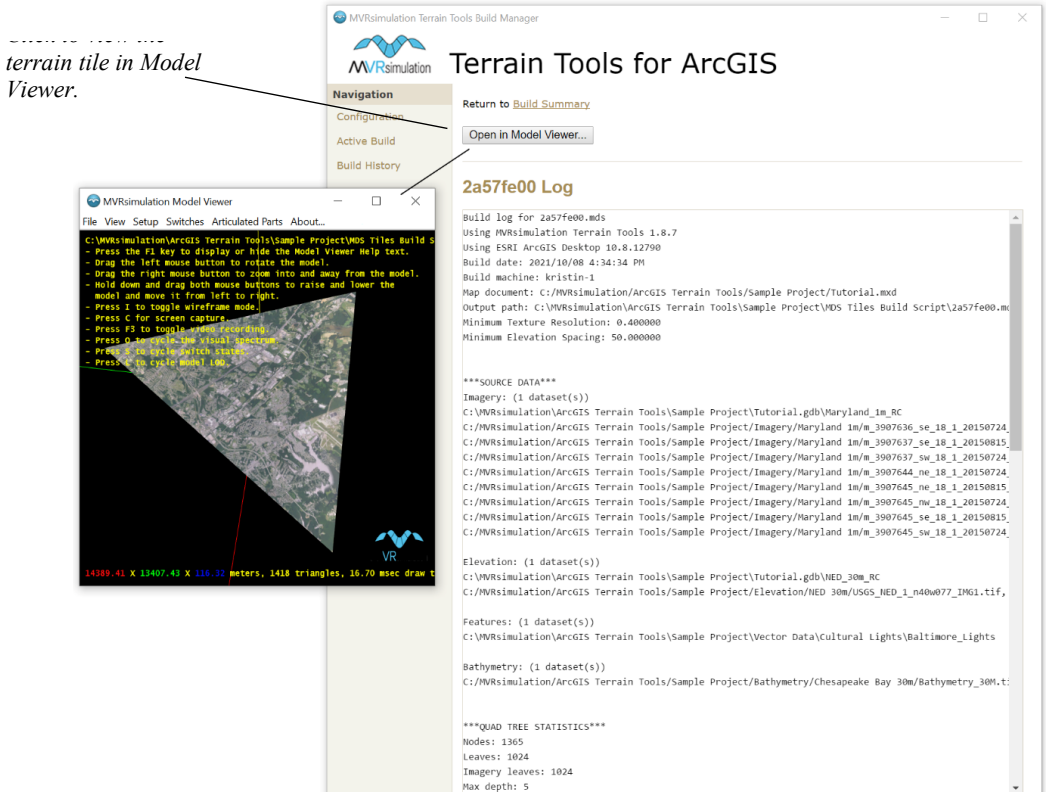
Click the Value field to change the minimum texture resolution and/or minimum elevation spacing for the selected tile(s) to rebuild.

Click the checkbox that corresponds to the tile you want to rebuild.

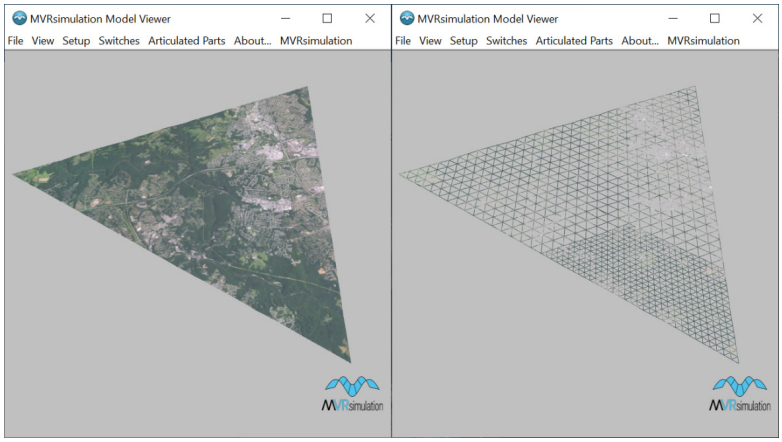
Inspecting a completed tile

New in Terrain Tools v2.0 is the ability to immediately inspect a tile in Model Viewer without leaving Terrain Tools. (You can also double-click on the tile's .mds file in Windows Explorer to view the tile in Model Viewer.)

On the log page of a selected tile, click Open in Model Viewer.

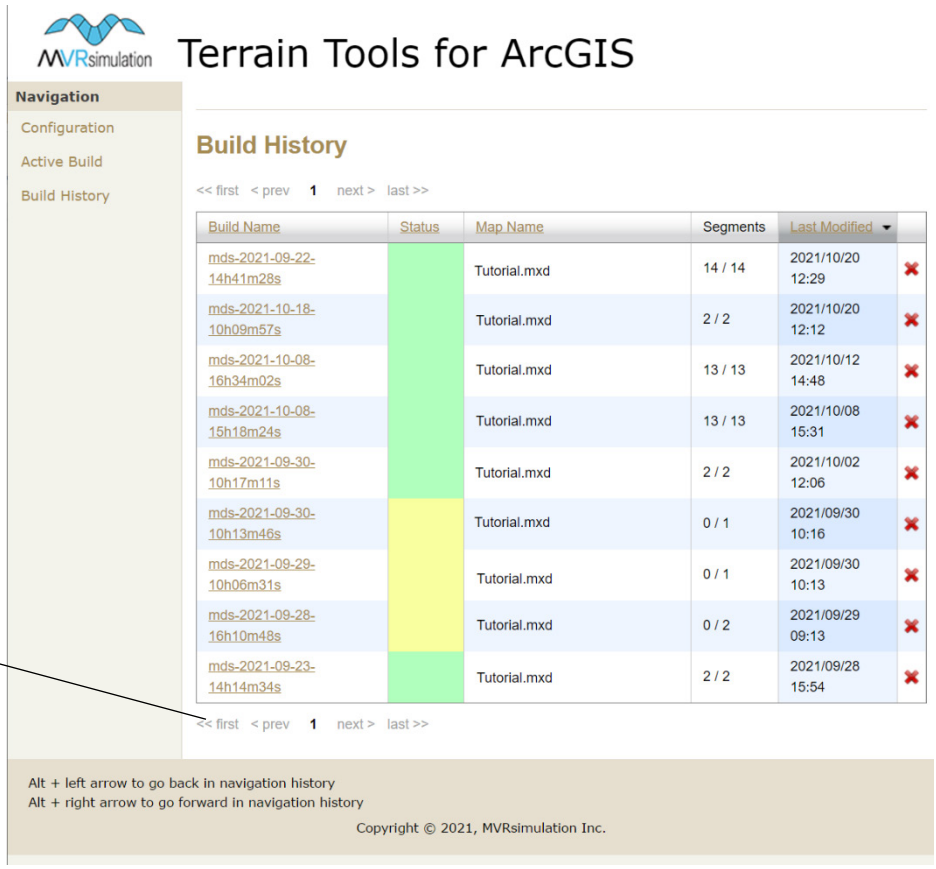


In Model Viewer, you can inspect the tile (without culture) with all the tools available for examining models, such as wireframe mode. (Press F1 to hide the onscreen text.)



Navigating multiple pages of Build History.

To speed up browsing through multiple pages of Build History, press the Alt and left arrow key to go back one page or the Alt and right arrow keys to advance forward one page. This is useful for quickly getting to the build history of a previously built tile to examine the settings or metadata.



Click to scroll through the build history.

Configuring machines to build terrain

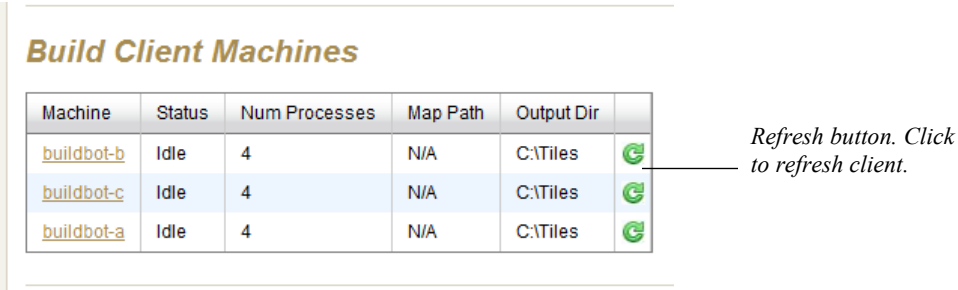
When Terrain Tools is first installed, the Build Manager is set up in a single machine configuration. That means the server and client are both running on the local machine. For single machine configurations, no further setup is necessary. For other configurations, such as a setup for building terrain on multiple machines or using Esri ArcEngine, the setup is slightly more complicated. The following section will discuss such scenarios.

Adding client machines

Simply connecting additional machines running on your local network is straightforward. If you are only using the client machine to build terrain, and will not be editing or configuring any data in ArcMap, no additional MVRsimulation Terrain Tools license or dongle is required.

To connect an additional client machine:

1. Install Esri ArcGIS Desktop or ArcEngine, and then install Terrain Tools on the remote machine.
2. Make sure the Build Manager is running on the machine (its icon should be visible in the system tray). If the Build Manager is not running, start it from the Windows Start menu.
3. Open the Build Manager window by double-clicking on the MVRsimulation icon on the system tray. Alternatively, right-click the icon and select Open Browser.
4. On the Server Configuration page, notice the Build Client Machines table, which will have one entry initially for the client process on this machine. Click the machine name link in the Machine column to load the preferences editor for that machine.
5. In the Preferences table, find the row which contains the Name buildServerAddress, and click on the Value field for that row to display the editor dialog box.
6. Enter the hostname or IP address of the *server* machine, and click Save.
7. If you previously changed the buildServerPort or webServerPort values on the server machine, make sure that both values match the client preferences.
8. *Note:* To connect the client to the new remote server, you *must* click the Refresh button at the end of the table row for the client. Alternatively, you can disable, and then enable the client from the right-click menu available from the Build Manager icon in the system tray on the client.



After performing either of these steps, the client should appear in the Build Client Machines table on the Configuration Page for the server. To open the Build Manager window for the newly connected server from the client machine, close, and then re-open the Build Manager window.

9. (*Optional*) Disable the server on the client machine(s). It is harmless to keep the server running on the client machine, even when the client is connected to a remote server; this step is optional.

Using map documents on client machines

The more error-prone step is ensuring that each machine has access to a properly configured map to build from. This can be further complicated on machines with only ArcEngine installed since ArcMap and ArcCatalog are not available to validate map (.mxd) documents and source data.

The two main challenges to resolve are:

- The Build Manager on the client machine needs to be able to locate the map corresponding to the active build.
- All data source mappings must be valid.

To allow Build Manager to locate the map, the map must have the same filename as the map in the original build request. It can be in any directory; however, if the client path differs from the build request path, the remote path must be specified in the mapSearchPath parameter on the client machine. For example, if the build references a map located on machine 1 at C:\Maps and a build client has the map on its D drive, D:\Maps, then D:\Maps must be present in the mapSearchPath list parameter on the client machine.

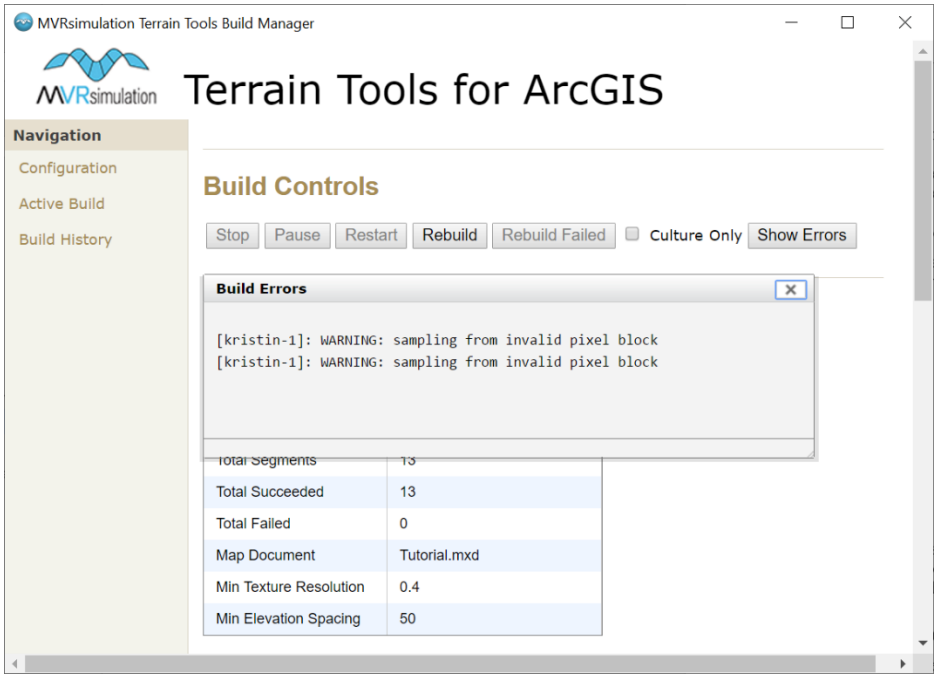
Validating data sources on remote maps requires manual intervention by the user. There are several approaches to ensuring maps built on one machine will work on a remote machine:

- Make all source data map to network locations visible from all client machines using one or more of the following:
  - Use the same network drive mapping on each machine that points to the shared data (i.e. map drive letter P to some shared network drive, then edit data sources in the map to P:\Source Data\...).
  - Use UNC paths to reference source data. (i.e. \\MACHINE-NAME\Source Data\...).
- Mirror all source data across the client machines using rsync, BeyondCompare, revision control software, or a similar application. In the map documents, either reference the mirrored local source data using relative paths or make sure the absolute path and drive letter match on all machines.

Using local imagery and elevation data can provide performance advantages but it makes configuring the map more difficult. You should try both approaches and choose the workflow which best suits your needs.

Troubleshooting

Terrain Tools has several resources integrated into the Build Manager window to aid with troubleshooting build warnings or errors. Logging output is captured from both the Build Manager as well as for each compiled terrain segment. When an error or issue is encountered in a terrain segment build process, an error or warning message is immediately displayed in a dialog box on the Active Build page:

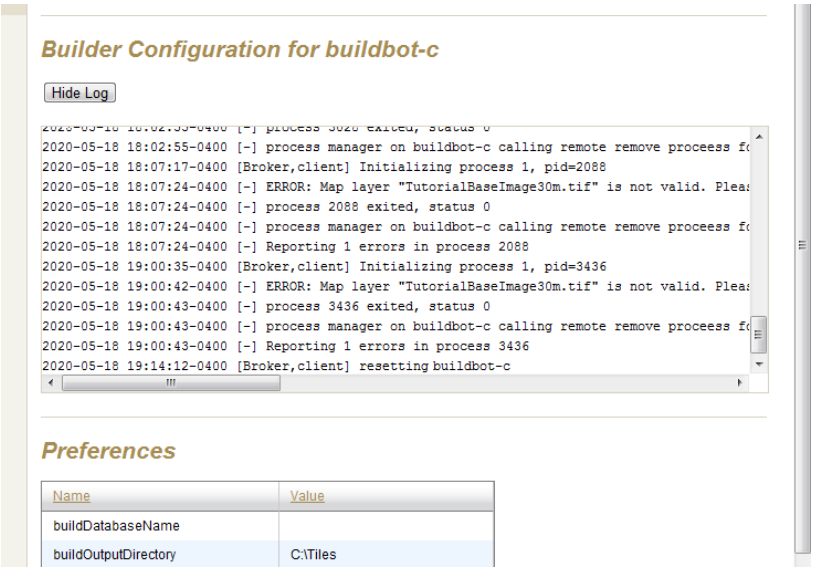


Some classes of errors will invalidate all subsequent builds on that machine. For instance, if a data source is invalid in the .mxd document, then all subsequent builds would fail. The client machine is marked as “Invalid” and excluded from the Build Manager until the error is fixed. This invalid state is encoded in the Build Client Machines table on the Configuration page.

To reinstate the client machine on the active build after you have resolved the problem, do one of the following:

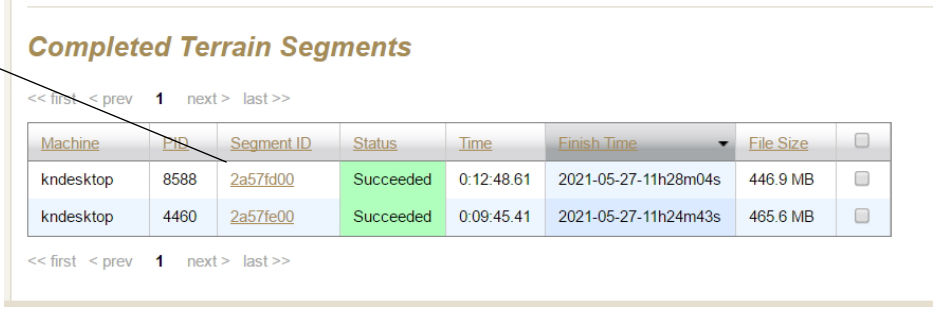
- Click the refresh button in the row of the build client that had the error.
- Manually disable, and then re-enable the client from the right-click menu available from the Build Manager icon in the system tray of the client machine.

All error output from terrain segment build processes is also appended to the client log, which is visible on the Client Configuration page. The client and server logs can be useful for troubleshooting problems with the Build Manager.

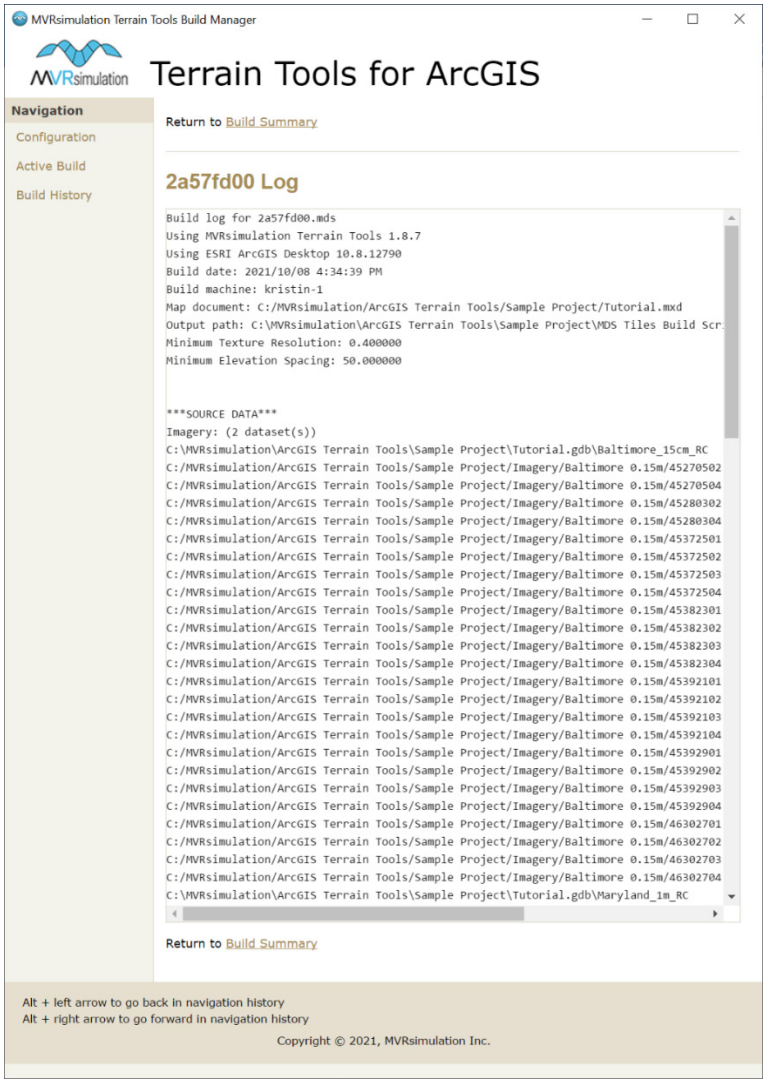


If the loggingVerbosity preference on the client machine is set to a value of '3' or higher, much of the compilation processing is logged in the terrain segment logs as well. To view the segment log, click the Segment ID link on either Active or Completed Terrain Segments tables.

Click to display the log for that segment.



This action opens the log page for the given terrain tile (segment), as shown in the following example for terrain tile 2a57fd00:



If you cannot determine the source of a problem, copy the output from the client/server logs and/or the terrain segment process from the Build Manager window and email the output and a description of your problem to support@mvrsimulation.com.

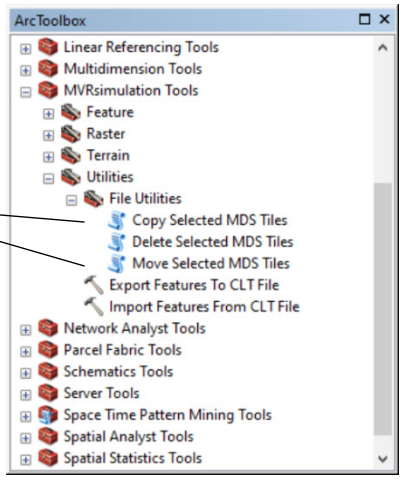
Copying, moving, or deleting completed tiles

Terrain Tools has tools for copying, moving, or deleting selected completed tiles from within ArcMap. These tools make it easy to manipulate tiles after they have been built, by selecting them on the map where you can see the geographic regions of the tiles you want to manipulate, rather than selecting them by their filenames in Windows Explorer.

To move, copy, or delete a set of tiles:

1. On the map, select the set of tiles you want to work with. (You might need to make the VRSG WorldTiles layer the only selectable layer.)
2. Choose the appropriate tool in the File Utilities toolset in the MVRsimulation Toolbox.

Tools for manipulating the tiles that are selected on the map.

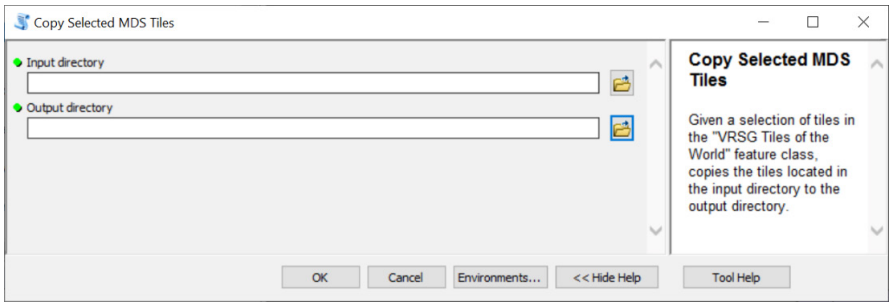


3. When the dialog box for the tool appears, browse to the source directory of terrain tiles you want to copy, move or delete.

In the case of copying or moving tiles, also browse to the target directory you want to copy or move the tiles to. (You might need to make an ArcGIS connection to the output/target directory.)

4. Click OK.

The following dialog box is a tool for copying terrain tiles to a specified directory:



Backward compatibility with older VRSG versions

By default, TerrainTools assumes compiled terrain will be viewed using the current version of VRSG. Over time, improvements might be introduced that are not backward compatible with legacy versions of VRSG (for example LOD morphing requires VRSG v6.3.401 or higher to load).

Terrain Tools v2.0 introduces a simple way to compile terrain that is compatible with a specific version of VRSG, by using the INI preference `targetVRSGVersion`. Terrain features and capabilities that are not compatible with the given version will be disabled in the compiled terrain.

For example, to create terrain compatible with rendering in VRSG v6.3.1, open \MVRsimulation\ArcGIS Terrain Tools\Config\TerrainTools_ <machine_name>_user.ini file in Notepad or another text editor. Add the following line and save the file:

```
targetVRSGVersion = 6.3.1
```

With that setting, newly compiled terrain would disable LOD morphing, colorization of instanced geometry, larger pageable textures, and several other more recent node structures.

The following table lists terrain improvements and the minimum VRSG version required to view them:

Terrain feature or capability	Minimum VRSG version
Pageable textures larger than 512 x 512	6.6.73
Extended HPX format support	6.4.157
Instancing colorization	6.4.157
GeometryMT node support in instancing	6.4.45
Improved LOD support for instancing clusters in viewports with narrow FOV	6.4.19
LOD morphing	6.3.401
Compiled in runways and insets	6.2.232
Instancing, CityEngine models, and forests	6.2.112
Microtextures	6.1.57

CHAPTER 6

Visualizing 3D Terrain in VRSG

Once you have built one or more terrain tiles, you can render and interact with the terrain in VRSG. In addition, with VRSG Scenario Editor, you can add cultural features to the terrain and create pattern-of-life scenarios with moving models using tools and content libraries with which you can build dense 3D scenes with realistic visual characteristics.

This chapter summarizes a few tasks to get you started visualizing terrain tiles in VRSG. You can find complete information about VRSG in the *MVRsimulation VRSG User's Guide* and Scenario Editor in the *MVRsimulation Scenario Editor User's Guide*.

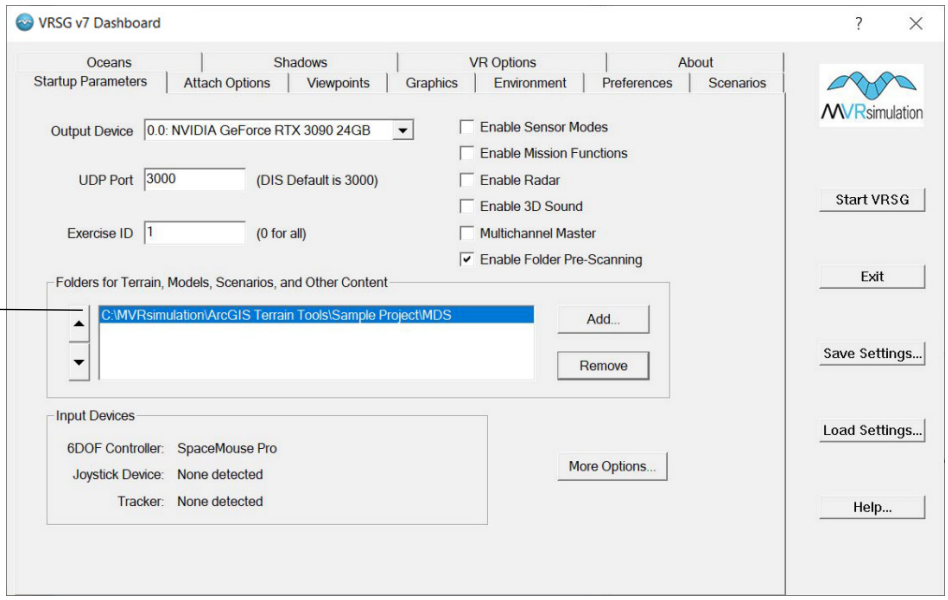
To visualize terrain built with Terrain Tools version 2.0 you should use VRSG version 6.0 or higher. To take advantage of Terrain Tools' support of the latest features in VRSG, VRSG version 7 is recommended.

Before you begin:

- Determine the location(s) of the terrain tiles you want to visualize. They might be located on multiple drives and directories.
- Copy any textures used for features in the terrain tiles, including any microtextures to a subdirectory called \Textures, located within a directory of tiles. For example to visualize the tutorial example, copy the texture files you used for road, buildings, fences, microtextures, and so on, from the directory in which the texture resides to a \Textures subdirectory within the directory that contains the tutorial terrain tiles you built.

To visualize terrain tiles in VRSG:

1. From the MVRsimulation program folder on the Start menu, choose VRSG 7.



Add the path(s) of your terrain tiles and textures used by features.

2. On the VRSG Dashboard's Startup Parameters tab, specify the directory paths of the terrain tiles and textures, and any scenarios that take place on the terrain that you want to visualize.

   The order in which the paths are listed is the order in which VRSG will search for content, starting with the path listed at the top. The search order becomes very important in cases where you have several files, each with the same name, in the different directories. VRSG will use the file listed in the first directory path it encounters (the top of the list).

   Note: As mentioned earlier, any textures used for features or microtextures must be in VRSG's search path, in a \Textures subdirectory of a terrain tiles directory you specified. You do not need to explicitly specify the \Textures subdirectory. VRSG will load all textures it finds in the subdirectories of the tile directories. Any vrsg.clt cultural feature file must also be in the VRSG search path. If you worked through the tutorial, a vrsg.clt file will be located in the \MDS directory. If any models with light lobes were compiled into the terrain, companion LLF files will reside in this directory as well.

   You can also use a VRSGTerrainSearchPath.txt file to specify search path directories, as described later in this chapter.

3. After you have specified the tile directory path(s), click the Start VRSG button to launch the visualization window.

The 3D terrain tiles appear in the VRSG visualization window, as in the following example from the tutorial terrain:

If only sky, not terrain, is visible in VRSG, it means that no default viewpoint for the terrain has been set yet. Drag a terrain tile file from the Windows File Explorer to the VRSG window to move the eyepoint to a location on the rendered terrain. Because coordinates are encoded within a tile's name, VRSG will automatically move the eyepoint to the center of that tile.

Once you get a view that you want to save to return to, set it as the default viewpoint by pressing the "S" key on the keyboard. Then, on the Viewpoints tab of the VRSG Dashboard, rename the viewpoint to "Default" so that the next time you load this terrain in VRSG, the terrain will load with the location of the viewpoint displayed in VRSG. Setting a viewpoint is described further later in this chapter.

Specifying terrain tile directories

In VRSG, you can visualize a set of terrain tiles located on multiple CPUs, drives, and directories. The order in which you list the paths is critical because it determines the order in which VRSG will search for tiles to render. For example, if you have a set of 1-meter resolution tiles stored in one directory, and have a smaller set of tiles at 0.50 meter resolution of the same geographic area stored in another directory, you would want to list the directory of high-resolution tiles first in VRSG's search list. To move a directory path higher or lower in this list, highlight the path and move it with the up- or down arrow key.

You do not have to visualize an entire set of tiles of an area; you can visualize any set of tiles. For example, you can place a subset of generated tiles in a different directory from the entire set, and point VRSG to the location only of the subset. VRSG automatically fills in areas that are not covered by terrain with tiles that use a water texture.

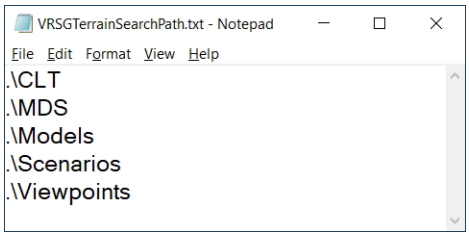
Because search paths for terrain tiles can be complex (residing on multiple drives, using multiple cultural feature files, textures used by features, microtextures, and so on) you can list a high-level directory in the Search Paths list, and then in that directory, provide a list of the

subdirectories you want VRSG to search. The list must be provided in a text file called VRSGTerrainSearchPath.txt.

This method of pointing VRSG to one directory and have other subdirectories added implicitly to the search path is useful in situations where:

- The underlying directory structure can change. In such cases, you do not have to keep up with such changes within VRSG itself; you simply edit the VRSGTerrainSearchPath.txt file.
- Large sets of terrain tiles need to be loaded by VRSG. You point to a single directory, and that directory can use VRSGTerrainSearchPath.txt to expand a complex list of directories and subdirectories.
- Multiple directories of cultural feature files (vrsg.clt files) and models.

An example of how VRSGTerrainSearchPath.txt is set up can be found in the VRSG installation: \MVRsimulation\VRSG\Terrain\Syria\Hajin\VRSGTerrainSearchPath.txt.



This search path text file lists all the \Hajin subdirectories that VRSG should search to render the Hajin terrain tiles, models, and scenarios that are delivered with VRSG.

The VRSGTerrainSearchPath.txt file can only store relative paths to directories; it cannot accept absolute paths. Each line must start with ".\". You can specify parallel subdirectories in the directory tree branching from another subdirectory, but be sure to include the initial ".\" before adding the backslash ("..\"). For example:

```
.\..\..\..\New Terrain Directory\MDS
```

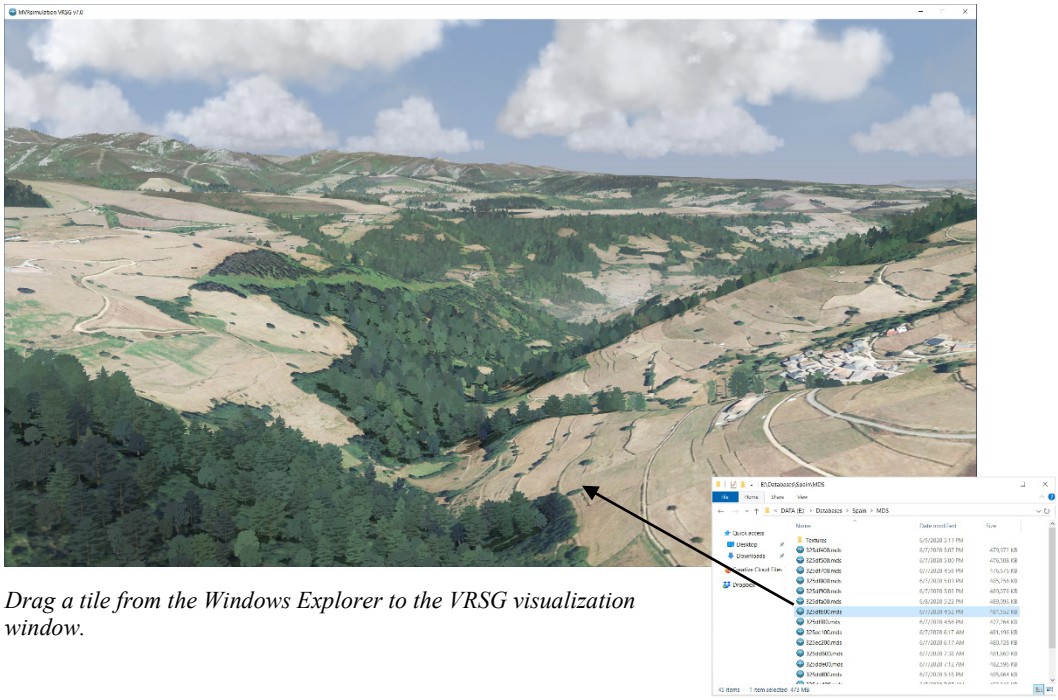
In the case of shared network drives, you can use the VRSGTerrainSearchPath.txt file to point to one directory per network drive. If tiles are stored on multiple shared network drives, you could specify the path to each drive on the Startup Parameters tab, and then add one or more VRSGTerrainSearchPath.txt files to each drive.

Setting viewpoints

When the VRSG visualization window opens, the default VRSG view will probably be nowhere near your tiles of interest; you are likely to simply see water or the default viewpoint of some tiles of another area you have been working within VRSG.

To move the eyepoint in the visualization window to one of your tiles, do one of the following:

- Enter the tile name (without extension) in the Teleport field of VRSG Dashboard's Viewpoints tab.
- Drag the MDS tile of interest from the Windows Explorer to the VRSG visualization window, as shown in the following example:



Drag a tile from the Windows Explorer to the VRSG visualization window.

This action moves the VRSG eyepoint to the center of that terrain tile.

You can save a viewpoint of this location so that you can return to it again, as described in the *MVRsimulation VRSG User's Guide.*

Placing cultural features on the terrain

Once the terrain tiles have been built and loaded into VRSG, you can optionally add cultural features to the terrain and create pattern-of-life scenarios that take place on the terrain.

- To simply place one or two cultural features on the terrain, drag a 3D model (in MVRsimulation's HPY or HPX model format) from Windows Explorer and drop it in the VRSG visualization window at a specific location on the terrain. To refine the placement of the model, attach to the model, and then move it or reorient it. Save the edit to a VRSG cultural feature file by pressing the J key on the keyboard. For more information about cultural feature files, see the *MVRsimulation VRSG User's Guide*.
- To place a large inset model, such as one generated by Esri CityEngine, you would edit create or edit the cultural future file (.clt) for the terrain to add the coordinates and

directory path of the model, as described in the appendix, "Adding CityEngine Models to 3D Terrain."

- To build up substantial content on the terrain, use the VRSG Scenario Editor application, which is installed with VRSG. This application provides a graphical interface with tools and content libraries you can use to build dense 3D scenes on your 3D terrain with realistic visual characteristics and to create pattern-of-life scenarios. For more information about adding cultural features to the terrain with Scenario Editor, see the *MVRsimulation VRSG Scenario Editor User's Guide*.

Using custom models

Culture models can be from MVRsimulation's large 3D content libraries, which are delivered with VRSG, or your own models that have been converted to MVRsimulation's model format. You can convert models from FBX or FLT format to MVRsimulation's format using the conversion utilities that are installed with VRSG and located in \MVRsimulation\Common\Util\.

To make your own custom models (in MVRsimulation's model format) available in VRSG, add the applicable model directory path(s) to VRSG's search path. To make your own models available in the Scenario Editor feature palettes, add the applicable model directory path(s) to Scenario Editor's search path.

Note: You can also supply thumbnail image files associated with your own custom models or effects to be displayed in the Scenario Editor feature palettes. Simply place a small square image of your model (in .jpg, static .gif, or .png format) in the same directory as your model or effect, with the same name as the model or effect. The image could be a screen capture taken in MVRsimulation's Model Viewer. When Scenario Editor finds that image in the same directory as the model or effect, it will display the image in the feature palettes.

Providing an overall terrain microtexture in VRSG

As mentioned in the chapter "Working with Terrain Features," in addition to supplying a microtexture for an area defined by a shapefile, you can also provide an overall microtexture in the terrain tiles directory that VRSG will render in the 3D scene at runtime. In fact, each directory of terrain tiles can have its own unique microtexture.

Generic textures for sand, grass, dirt, and gravel are examples of geotypical microtextures that can blend with the geospecific base terrain imagery to produce the most realistic visual depiction of the virtual world.

The texture can be in any image format that VRSG supports, such as JPG, BMP, TIF, and so on. (The full list of VRSG-supported image formats can be found in the chapter "Working with Terrain Features.") The microtexture must be located the \Textures subdirectory of the terrain tiles directory and must have a name in the form:

```
vrsg_microtexture_ X_Y_Z.tex
```

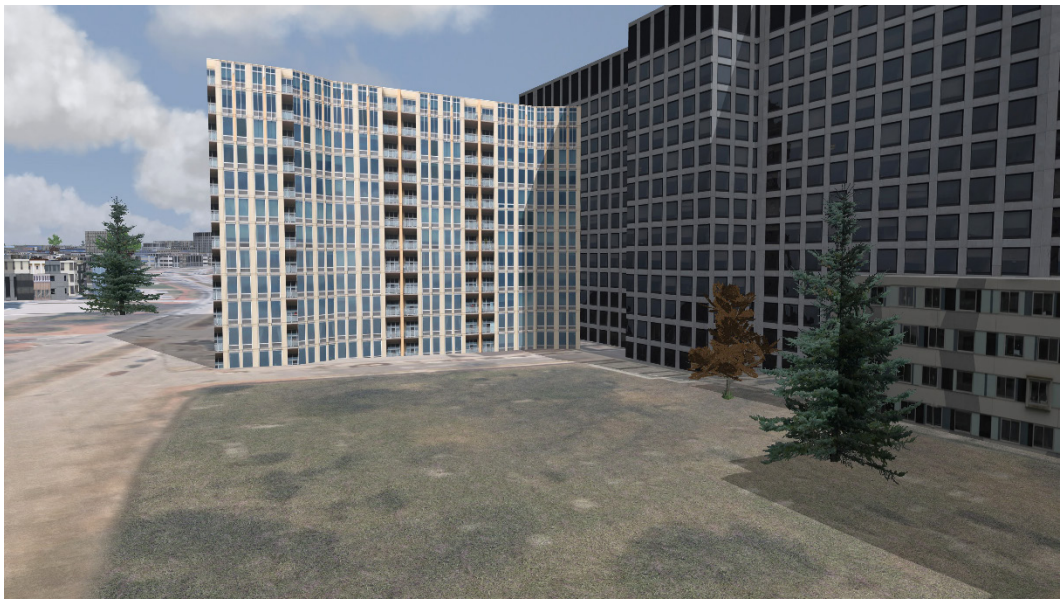
where X, Y, and Z are parameters that control the microtexture mapping.

- X – is the mapping frequency, in units of texture repetitions per meter. A value of 0.1 means that the microtexture pattern covers an area of 10 x 10 meters.
- Y – is the fade-out distance in meters. This is the distance at which the microtexture disappears completely.
- Z – is the blend bias. This value is the minimum percentage of base texture that shows through at close range.

VRSG will apply that texture to tiles in that directory. For example:

```
vrsg_microtexture_ 0.1_800_0.15.tex
```

The following example shows overall dirt microtexture rendered by VRSG at runtime on the tutorial sample project terrain. The example also shows the dirt microtexture blending with a grass microtexture specific to an area, placed via a shapefile and compiled in the terrain in Terrain Tools:



For VRSG to render the 3D scene in sensor mode, you can also provide an IR alternative of the microtexture. VRSG will look for an IR texture of the form:

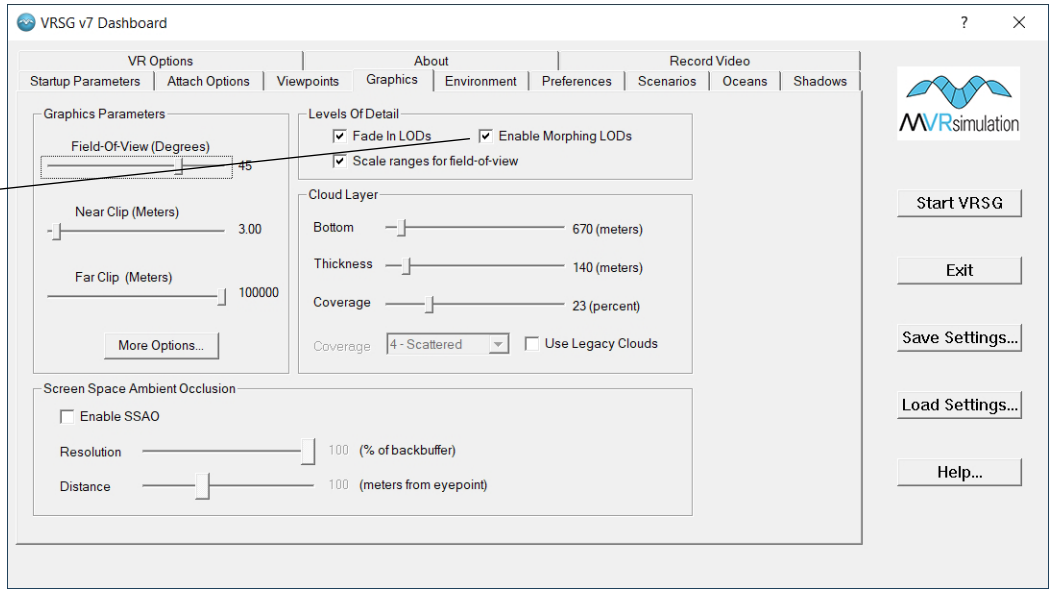
```
vrsg_microtexture_ X_Y_Z_IR.tex
```

Rendering the terrain LODs of legacy terrain tiles

As described in the chapter “Refining Terrain Construction,” Terrain Tools(supported by VRSG version 6.4 and higher) introduces an improvement for rendering the transition LODs. For terrain compiled in this new technique, LOD morphing, terrain vertices are blended from their true elevation towards the elevation of the next lower LOD before the lower LOD is reached. This technique replaces the legacy LOD fade-blending method compiled in terrain in previous versions of Terrain Tools.

By default, VRSG (version 6.4 and higher) renders terrain with the LOD morphing technique. If you need VRSG to render older terrain (created in Terrain Tools version 1.5 or older) or a mix of older terrain and new terrain created with Terrain Tools 1.6 and higher, unselect the Enable Morphing LODs checkbox on the VRSG Dashboard's Graphics tab. This action will force VRSG to render all terrain with the legacy LOD fade-blending method:

Unselect this option to have VRSG render the terrain with the legacy LOD fade-blending method.



Creating VRSG scenarios on the terrain

VRSG Scenario Editor, which is installed with VRSG, enables you to create and edit real-time 3D scenarios on your 3D terrain, to playback in VRSG.

Experienced VRSG users and novices alike can work in a flexible manner to increase the realism of terrain easily with rich culture and scripted movements of vehicles and characters. The 3D terrain that you work with in Scenario Editor is the same 3D terrain you visualize in VRSG, and the scenarios you create can be run in both Scenario Editor and VRSG.

Many aspects of Scenario Editor are familiar from VRSG, such as creating and using viewpoints, adding static cultural and dynamic models, effects, appearances, cultural feature files, manipulating characters with weapons and animations, and content search paths.

Because Scenario Editor contains a 3D terrain workspace, you see the exact same terrain you created in Terrain Tools and would see in VRSG. This means you can work with accuracy and precision in placing and refining your static culture and in scripting the movements of vehicles and characters, in environments such as dense urban scenes, airfields, and forward operating bases.

Changes you make in Scenario Editor can be put to immediate use in the VRSG simulated training environment. With Scenario Editor, you can easily update training areas of interest on the terrain across shared scenarios or just for a specific scenario.

Note: When you export a scenario for use in VRSG, Scenario Editor exports the footprints of buildings as polygon shapefiles. The .shp, .dbf and .shx files are exported in the standard shapefile format and can be imported into your map. The shapefile contains bounding boxes in geocentric coordinates of the position (but not the orientation) of each culture model.

To become acquainted with Scenario Editor's capabilities, you can run demo scenarios that are installed with VRSG. You can also watch real-time video recordings of the demo scenarios on the MVRsimulation website at: www.mvrsimulation.com.

Examining terrain tiles in the Model Viewer

In addition to using the MVRsimulation Model Viewer for examining MVRsimulation models, you can use the viewer to quickly examine an individual terrain tile, for example, to inspect the tile's textures or geometry.

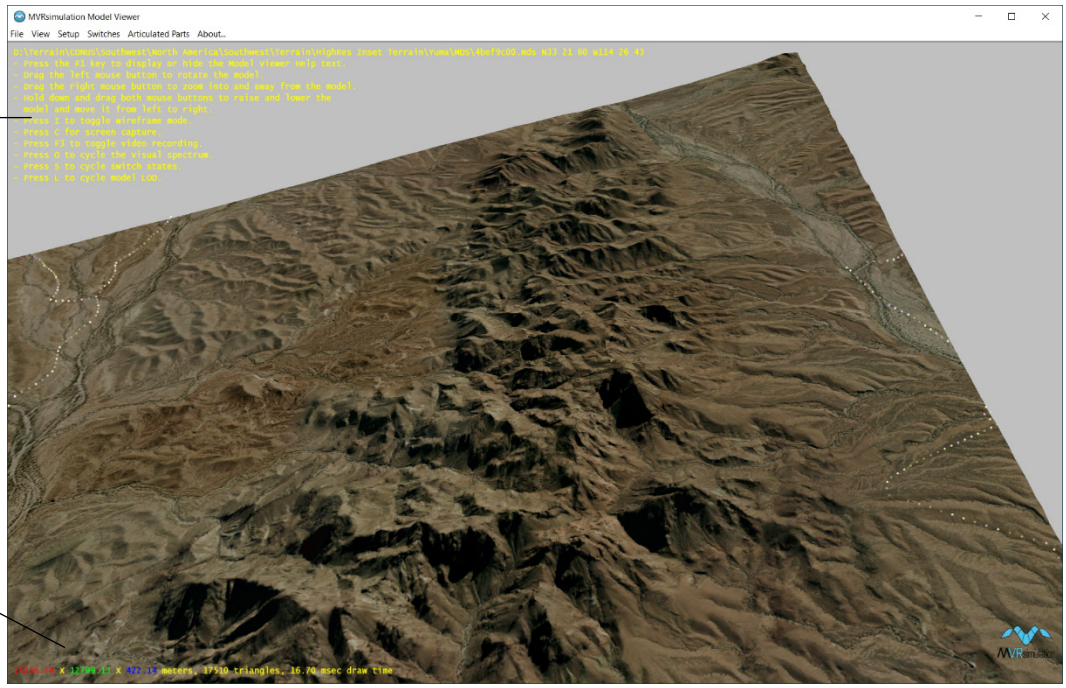
To view a terrain tile in the Model Viewer:

1. From the Windows Start menu, choose MVRsimulation Model Viewer.
2. When the Model Viewer appears, choose File > Open, and then from the Files of Type drop-down list, select MVRsimulation Terrain Tiles (mds).
3. Browse for the MDS tile you want to open and click OK.

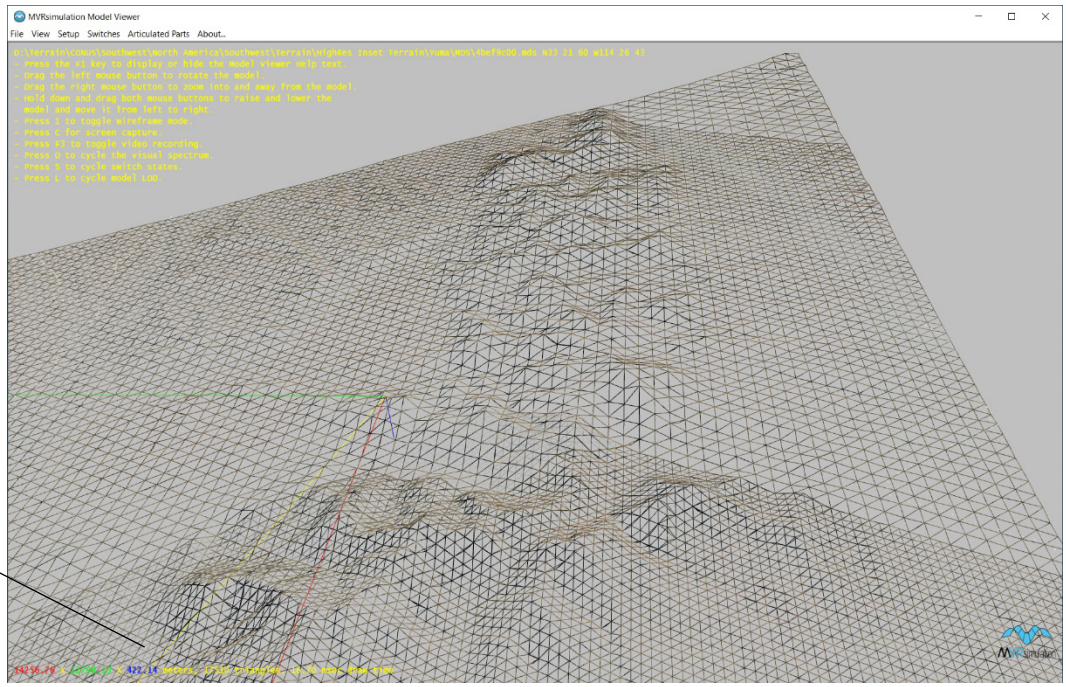
Alternatively, you can simply double-click a terrain tile (.mds file) in the Windows Explorer to open it in the Model Viewer, or if the Model Viewer is already open, you can drag an MDS tile file from the Windows Explorer to the Model Viewer to view it.

Onscreen help.

Geospecific and polygon information about the tile.



Once the terrain tile is open in the Model Viewer, you can zoom, rotate, and inspect the tile's textures and geometry as you would any other object in the Model Viewer. You can use the viewing options by pressing a specific key on the keyboard as listed in the onscreen Help text as shown above. For example, press the "I" key to display the tile in wireframe mode as shown next:



The same tile in wireframe mode.

Note that some viewing options in the Model Viewer apply only to models, not to terrain tiles. See the *MVRsimulation VRSG User's Guide* for more information about the Model Viewer.

CHAPTER 7

VRSG Terrain Architecture

MVRsimulation's round-earth VRSG terrain architecture is designed to solve a variety of problems associated with projection-based, monolithic visual databases. Through this architecture, improvements are realized in database production, distribution, storage, and update, as well as in many run-time and mission functions. Benefits include use of geocentric rather than projected coordinate systems, live update of terrain, scalable and robust terrain generation, and index-free demand paging.

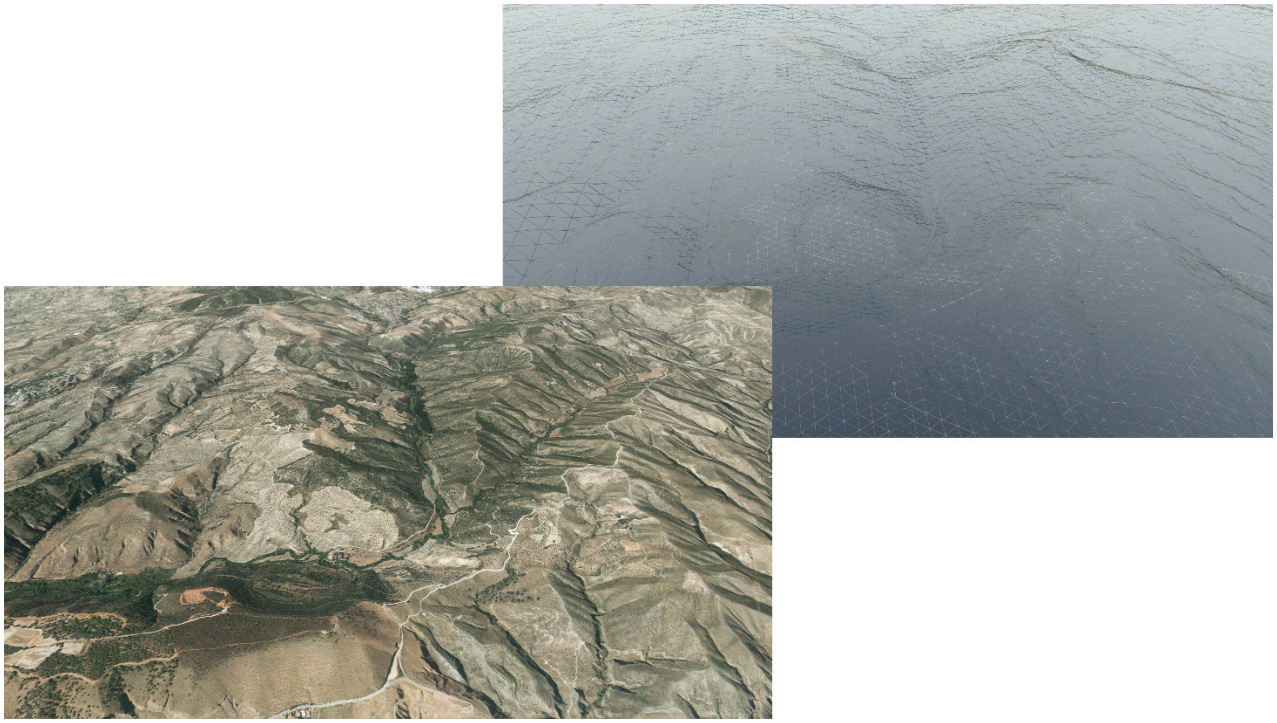
This chapter provides an in-depth look at the VRSG terrain architecture, including design and implementation details to help you plan future terrain projects more effectively.

VRSG real-time rendering of F/A-18 entities (from MVRsimulation's 3D content libraries) in flight over the geospecific 2 cm per-pixel resolution 3D terrain of a target range on the Fallon Range Training Complex at Naval Air Station (NAS) Fallon, NV. The 2 cm virtual terrain was built with 2 cm imagery collected by MVRsimulation's small UAS.

Background

Virtual terrain databases for three-dimensional visualization are typically based on some type of map projection. Some common map projections used are Universal Transverse Mercator (UTM), Geodetic, Lambert Conic, and Flat Earth among several others. A database projection

warps the curved surface of the earth onto a flat X-Y plane to create a local Cartesian coordinate system that simplifies the mathematics of a simulation.



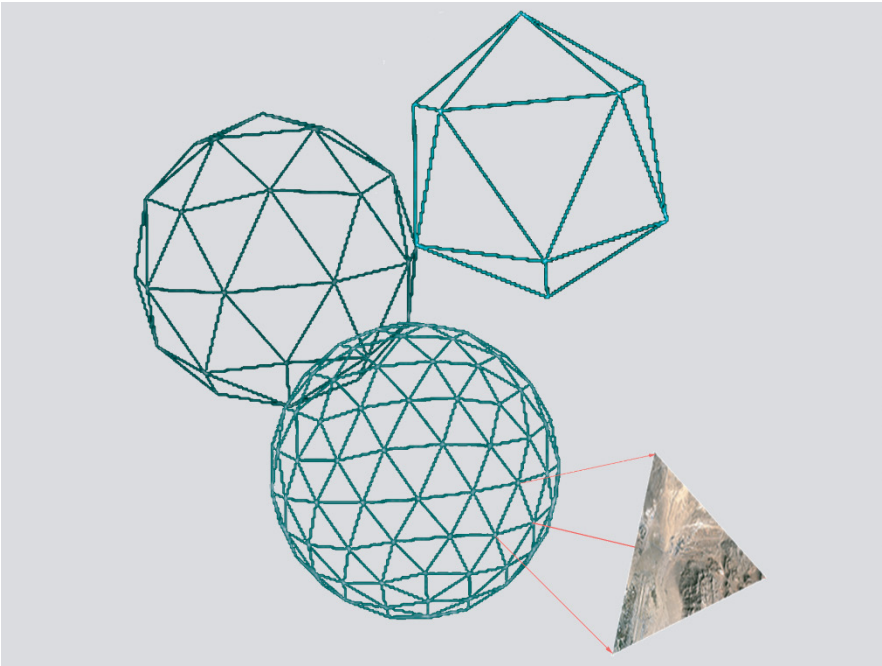
VRSG real-time and wire-frame views of MVRsimulation's geospecific 3D terrain of Granada, Spain.

A more ideal and accurate representation of the earth's surface is an earth-centered, earth-fixed coordinate system that accurately represents curvature of the earth and deals with ordinate axis convergence at the poles. Such a coordinate system is often referred to as geocentric. Projection-based database formats do not handle these conditions properly, creating a motivation for a geocentric database. However, most image generation systems have difficulty working in a geocentric coordinate system. Furthermore, since there is no XY plane onto which the database has been projected, classic tiling schemes used for database paging do not map well into the geocentric environment.

The following goals of the VRSG terrain format address all these shortcomings of the classic projection-based database architecture:

- A geocentric terrain representation to accurately represent earth curvature and handle polar regions correctly.
- A pageable format to enable databases of arbitrary resolution and geographic coverage.
- Support for sparse coverage where the areas to be included in the database need not be rectangular.
- A metadata-free design whereby the indexing overhead of the terrain data does not scale with geographic coverage or resolution of the source data.

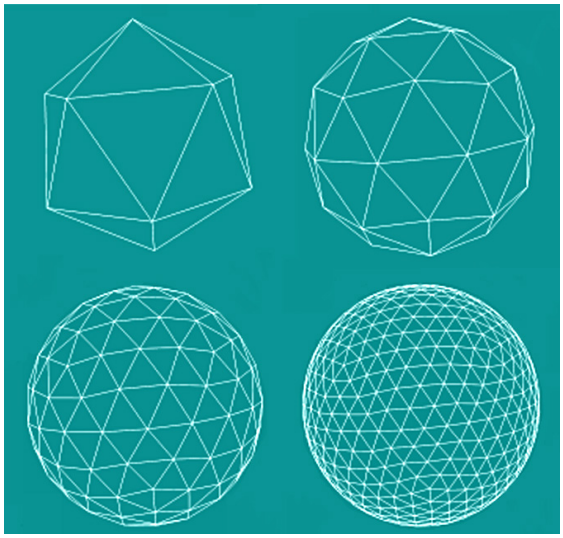
- The ability to update small regions of the database in short periods of time without the need to restart the run-time system.
- The ability to update the database incrementally, by adding to its coverage or updating pre-compiled regions with newer source data.
- A more robust database generation process that can be parallelized and restarted following computer or software failure.



Projection-based database architectures typically subdivide the XY extents of a desired geographic region into small, uniformly-sized, rectangular areas commonly referred to as *tiles* or *patches*. The geographic coverage of the entire database is then a mosaic of these tiles. Database tiles typically represent a unit of database compiler output, a load module or unit of paging by the runtime system, and/or a unit of culling by the runtime system. A key and enabling component to the VRSG terrain format is the use of a more general atomic geometric primitive as the database tile: the triangle.

In the VRSG terrain architecture, an ellipsoid model is subdivided into a set of triangles using what is known as a geodesic tessellation. A geodesic tessellation of an ellipsoid has the property that the triangles generated are nearly equal in area, and nearly equilateral in shape. A geodesic tessellation starts with a seed polyhedron consisting of a handful of triangles. Each triangle is subdivided into four by bisecting each of the three triangle edges. Upon bisection, the newly created vertices are then placed on the ellipsoid. This process continues recursively until the desired level of tessellation is achieved.

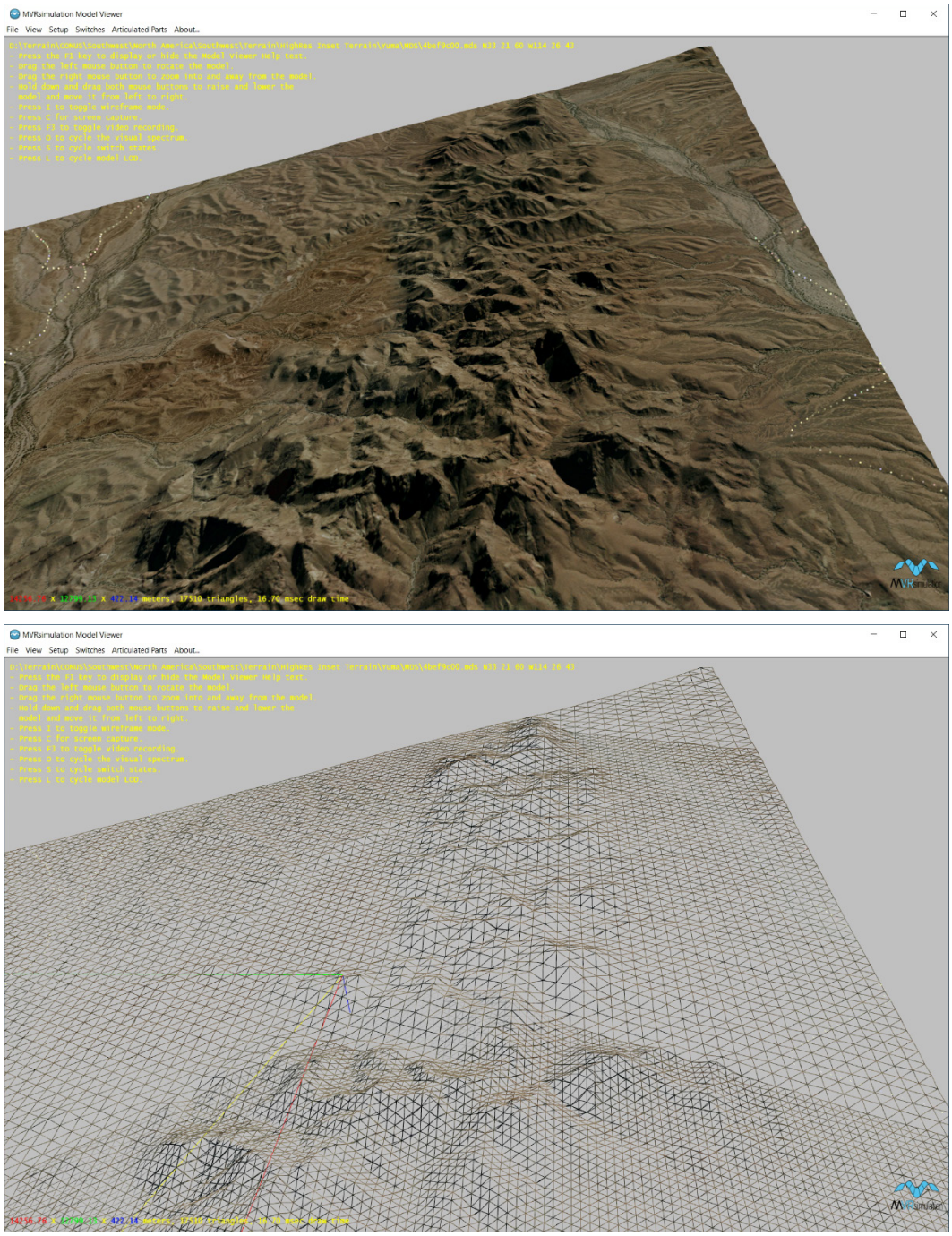
The following set of images illustrates how a seed polyhedron converges to a target ellipsoid after three levels of recursion in the geodesic tessellation process:



As an example, for a fixed-wing simulator application with a visibility range requirement of 200 km, the surface area of each database tile should be about 100 km^2. A 100 km^2 tile with a 200 km visibility range strikes an efficient balance between the number of addressable tiles in the scene, and the manageable unit of data to demand-page from disk in an atomic operation. To achieve a tile size of approximately 100 km^2, nine levels of recursion over the WGS84 ellipsoid are required, yielding 5,242,880 total triangles. Across the entire set of triangles covering the WGS84 ellipsoid, the areas of these triangles range from 90.0 km^2 to 117.3 km^2.

Each tile is represented on disk by a single file. The filename is a bit pattern that records the recursion history of the tessellation process leading to any given tile. This naming convention ensures that unique and repeatable filenames will be generated. It also negates the need for explicit indexing information. Eliminating such metadata from the database representation prevents overhead that would grow with geographic extent or increased resolution of source data. Thus, a metadata-free design is an important property that helps achieve this goal.

The following images show a VRSG terrain tile of the Prospect Square area of Yuma Proving Ground displayed in MVRsimulation's Model Viewer in normal texture and wireframe views. The 2 cm virtual terrain was built with 2 cm imagery collected by MVRsimulation's data collection small UAS:



Textured and wire-frame views of a terrain tile in MVRsimulation's Model Viewer.

Each tile is stored in a coordinate system that is a linear transformation of the Earth-Centered, Earth-Fixed (ECEF) reference ellipsoid. The geometry is stored in a local floating point coordinate system to reduce the dynamic range of working values. In this local transformation the Z axis is the tile's local surface normal and aligns with the ellipsoid normal. The local

origin is the center of the tile, and the X and Y axes lie in the plane of the tile's defining triangle. The orientation of the X and Y axes can be chosen arbitrarily and is optimized under other considerations. When rendered by the runtime system, each tile is rotated and translated into the current viewing coordinate system.

Implications for database compilation

For the database compiler, each tile represents a single unit of work. Thus the database can be updated or extended in 100 km^2 segments. The database compilation process only ever risks the time required to generate a single tile (on the order of minutes). Multiple database compilation processes work in parallel on independent tiles within the geographic region(s) of interest. In the event of a hardware or software failure, the database compilation process can be restarted, picking up where it left off. The only work lost is the time taken to produce the last partial tile at the point of failure.

Another important property is that a given database compilation process, working on a potential sub-region of the entire area, need only see the source data that is relevant to that sub-region. This enables the database generation process to begin before the entirety of source data has been received or pre-processed. This also allows source data to be spread across multiple computers, each computer contributing to a different sub-region of the entire area mitigating the need for very large single-volume storage such as a RAID.

The VRSG terrain compilation process also affords a more pipelined visualization process. In contrast, the monolithic database generation process was serial in nature, where the time to produce the database was the sum of the following subtasks:

1. Acquire source data.
2. Modify, correct, or enhance source data.
3. Compile the visual database.
4. Verify the visual database in the runtime system.

The VRSG terrain compilation process is fundamentally pipelined, as all four steps occur in sequence on time scales in the range of a few minutes. For example, as source material is acquired, it can immediately be modified, corrected, and enhanced for the compilation process before all remaining source material arrives. Whatever source material is available can be given to the compilation stage to begin producing tiles. When each tile is compiled, the result can be immediately validated in the runtime system without waiting for any other portion of the desired coverage area to be completed.

The VRSG terrain pipelined architecture is also highly scalable. Because each compilation is independent, database engineers can devote an arbitrary number of compilers to the task at hand. In practice, when as many as fifteen compilers operate on shared datasets of varying resolution and constituent source data, the overall performance improves nearly linearly with the number of compilers. MVRsimulation anticipates that for 1Gbps networks and commensurate storage media, this performance analysis would scale to as many as 100 compilers. As the number of compilers increases, network communication will become a bottleneck (either reading/copying source data, or writing/flagging completed tiles). With 10 Gbps networks becoming commonplace, this bottleneck is further loosened and the linear range will increase even further.

Implications for mission functions

Simulators often depend on the terrain database for such functions as elevation lookup and point-to-point intervisibility tests. Although the VRSG terrain architecture might appear on the surface to be difficult to support mission function features, it is not the case. Optimized terrain elevation lookups and intervisibility tests have been developed that exploit the recursive and hierarchal nature of the design and are very efficient. For example, the VRSG implementation of an elevation lookup is as fast as 20 microseconds, and point-to-point intervisibility lookups are as fast as 100 microseconds. These speeds are comparable to conventional, projection-based, terrain database implementations and in some cases even faster.

The advantages of the geocentric-based visualization architecture are obvious for point-to-point intervisibility tests. In particular, VRSG terrain handles long-range intervisibility tests correctly. Projected databases fail to produce accurate results under such circumstances because they cannot accurately account for the effect of the earth's curvature.

VRSG real-time rendering of a T-6C aircraft entity (from MVRsimulation's 3D content libraries) in flight over MVRsimulation's heavily forested geospecific 3D terrain of Lugo province, Spain.

Implications for threat systems

SAF applications such as OneSAF, JointSAF, XCITE, the U.S. Navy's Next-Generation Threat System (NGTS), and commercial systems such as BattleSpace Simulations' Modern Air Combat Environment (MACE) which can utilize VRSG terrain tiles for their elevation source can employ fully correlated databases without the need for ground-clamping. Databases derived from third-party products, such as Presagis Terra Vista, can also achieve full correlation provided the VRSG terrain tiles were produced from the OpenFlight output from the same Terra Vista project that rendered the SAF representation of the database. As a last resort, VRSG's ground-clamping feature can be used for cases where perfect correlation between the 3D database and the SAF database is not possible.

Implications for terrain scenarios

The VRSG terrain architecture provides a convenient mechanism for assembling custom terrain databases that draw together disparate geographic regions. Within the architecture the population of geodesic tiles may be sparse at run-time. Each tile is in fact an independent database. This sparsity can be exploited in situations where disparate training objectives need to be consolidated into a single scenario.

Consider an example scenario that includes ground attack, coastal overflight, surface-to-air missile evasion, and nap-of-the-earth terrain following. In conventional database architectures, a database design and supporting source data would necessarily have to be contrived. The development process would not be reusable and the resulting play-box would be geographically inefficient. Within the VRSG terrain architecture, specific and disperse geography can be brought together simply by addressing modular tiles built from source data of the desired regions.

APPENDIX A

Terrain Tools Preferences

This appendix lists the preferences available in Terrain Tools.

Build Manager preferences

Preference	Value
auxExternalTexturePath	Path of an additional \Textures directory for VRSG to search for textures for compiled terrain in addition to the default \Textures directory of the Terrain Tools or VRSG installation.
browserUpdateRate	Elapsed time in seconds between calls to refresh the dynamic data in the Build Manager window. (Default is 1 second.)
buildOutputDirectory	Output directory where compiled terrain is stored.
buildServerAddress	Hostname or IP address of the build server. (Default is 127.0.0.1, or localhost.)
buildServerPort	Port the build server uses. (Defaults to 8900.)
cultureFadeRange	Range, in meters, of LOD switch out distance at which cultural lights, fences, buildings fade out on the terrain. Default is 15,000 meters.
enableSegmentNeighbor Balancing	Enables the terrain segment neighbor balancing algorithm. (Default is true.)
loggingVerbosity	Controls the detail of logging information output for each tile. 0=disabled, 1=metadata only, 2=metadata, and performance stats, 3=metadata, stats, and verbose debugging output. (Default is 1.)
mapSearchPath	List of path entries (semicolon delimited), each of which is used to locate the ArcMap .mxd document for the active build.

Preference	Value
maxActiveBuildProcesses	Limits the number of simultaneously running build processes. (Default is -1, which lets the client determine the maximum number based on the number of processors on the machine.)
minElevationSpacing	Default minimum resolution of elevation used. *Note:* Only used to fill in defaults in the ArcToolbox tool parameters.
minTextureResolution	Default minimum resolution of imagery used. *Note:* Only used to fill in defaults in the ArcToolbox tool parameters.
modelSearchPath	List of path entries (semicolon delimited), each used to locate models for the active build that are not stored in MVRsimulation's model libraries (\MVRsimulation\VRSG\Models).
showSaveDocumentDialog	Enables the helper dialog box to remind users to save the map when initiating a build from within ArcMap. (Default is enabled/true.)
webServerPort	Port that the Build Manager window's data server runs on. (Default is 8901.)
targetVRSGVersion	A period-separated VRSG version number in the format [major].[minor].[build]. Use this preference to specify a particular VRSG version with which the output terrain tiles must be compatible. Features and capabilities in Terrain Tools that are not compatible with this specified version will be disabled in the output terrain. (Example: targetVRSGVersion = 6.2.1 would disable features such as LOD morphing, and JSON file support.)

Additional preferences

Terrain Tools also has a number of advanced preferences that are not available for edit in the Build Manager. Most of the time, the user will not need to change these from their default values. To override any of these preferences, edit the TerrainTools_[machine name]_user.ini file, located in the \MVRsimulation\ArcGIS Terrain Tools\Config\ directory. For user reference, the default values for these parameters are defined in TerrainTools_default.ini located in the same directory. When Terrain Tools is first installed, the _user.ini file is empty, meaning all preferences are set to their default values.

Preference	Value
sampleInstanceColor	Directs instanced vegetation models to sample their color from the source data in the Imagery group layer. (Default is true, enabled.)
sampleInstanceColorTol	A tolerance value that represents the maximum difference (in color space) between the sampled color and the average vegetation color in the source model. (Default is 250.)
overrideInstanceColorRGBA	A comma separated list of red, green, blue, alpha color components (0-255) used to modulate the sampled color for vegetation. An alpha value of 0 disables the color override. (Disabled by default.)
instancingCount	Controls the clustering characteristics for instanced geometry. This is the maximum number of instances in an instancing cluster. The cluster will be subdivided if this value is exceeded. (Default is 15000.)
instancingWidth	Controls the clustering characteristics for instanced geometry. This is the maximum width/height dimension in meters of an instancing cluster. (Default is 4000 meters.)
defaultBathymetrySpacing	The resolution in meters of the procedurally generated bathymetry raster (based on the distance to the shoreline). (Default is 100 meters.)
sample2DWaterZ	For performance reasons, 2D water does not sample elevation from source data by default. Set this preference to true to force 2D water vertices to sample elevations from the source data (note that doing so may result in water geometry that is flat).
defaultWaterZ	The default Z value for water features that are not Z-enabled. (Default is 0 meters.) This preference is ignored if sample2DWaterZ is true.
exportCLTVersion	The version of exported .clt files.
cutInBlendDistance	The distance in meters at which to blend cut-in features. To avoid the formation of cliffs in the terrain when cut-in features (such as roads, elevation features) are compiled, surrounding terrain vertices will blend the source sampled elevation with the cut-in feature elevation. Any terrain vertices that are closer to a feature than this value will feather blend the elevation value (modulated by the distance). (Default is 20 meters.)

Preference	Value
defaultCityEngineLODRanges	A comma-separated list of the maximum range CityEngine geometry LODs will be visible from, ordered highest detail to lowest detail. (Default is 1500, 3000, 5000, 7500, 10000.)
overrideRunwayVanishDistance	For cut-in runway/inset models, this value will override any vanish distance defined in the source model. (Defaults to 0, meaning the vanish distance in the model is preserved.)
modelSubfaceOffset	To avoid Z-fighting between base runway geometry and overlay layers when geometry is fit to elevation source data, overlay meshes with be offset by this amount (in meters). (Default is 0.05 meters.)
modelSubdivideElevRes	For runway meshes with multiple layers, this is the elevation resolution of a regular terrain mesh that will be cut into each runway mesh. This makes the layers more conformant with each other and less likely to Z-fight when fit to elevation source data. Default is 0, meaning the elevation resolution is derived from the local terrain that the model is cut into.
enableLODMorphing	Indicates whether LOD morphing is enabled. (Default is true/enabled.)

APPENDIX B

Read API for VRSG Terrain Tiles

MVRsimulation provides a free Read Application Programmer's Interface (API) for VRSG-formatted round-earth terrain tiles to support host-side mission functions. The API library is available for both Windows and Linux 64-bit systems.

- Windows 64-bit applications should link to MdsAPI.lib; MdsAPI.dll is required for the runtime.
- Linux applications should link to libMdsRead_x64.a. The Linux library was built with gcc version 2.96 20000731 (Red Hat Linux 7.3 2.96-110).

MVRsimulation customers on active maintenance can download the latest MVRsimulation Read API for VRSG terrain tiles from appropriate Windows or Linux subdirectories in the /Software/MdsAPI section of MVRsimulation's Download Server. Customers on active maintenance who need an account on the Download Server can request an account by sending a request to downloads@mvrsimulation.com.

Using the API library with multi-threaded applications

For the MdsAPI library to work with multi-threaded applications, ensure the setup calls are executed in a single thread, as follows:

1. Before any multi-threaded access, call mdsInitialize(), mdsAddDirectoryToSearchPath() to initialize the API.
2. After initialization, use as many threads as needed to call mdsGetElevation().

Note: The first time a terrain tile is loaded, it might take some time for the API to fetch the tile from disk and unpack the data. The thread that invokes loading the tile will block other threads from accessing the API while the shared data structures are being updated.

APPENDIX C

Adding CityEngine Generated Models to 3D Terrain

You can use Esri CityEngine to generate a 3D urban model of a large number of buildings to supplement terrain compiled in Terrain Tools, for rendering in VRSG. Terrain Tools supports the use of CityEngine rules package files (RPKs) for adding large areas of dense culture to 3D terrain quickly and easily.

If you have CityEngine, you can also create cultural content with this simple workflow, where you output the urban model from CityEngine to FBX format or Collada's DAE format (for converting the latter to FLT format in Presagis Creator), and then use MVRsimulation's FBX or FLT model conversion utility to convert the model to MVRsimulation's model format. Finally, you reference the urban model in a cultural feature file (vrsg.clt) like any other model that you want VRSG to load at runtime.

Left: The city of Hajin, Syria's building footprints in Esri CityEngine prior to exporting an urban model of 13,326 realistic 3D buildings, in FBX format for conversion to MVRsimulation's model format.

Below: The resulting city rendered in VRSG. The terrain was built with 50 cm imagery in MVRsimulation Terrain Tools for ArcGIS.

CityEngine generates procedurally created models from building footprints and road vectors using Open Street Map (OSM) data and your provided elevation or heightmap data. These building footprints and road vectors are extruded and textured from OSM data using Computer Generated Architecture (CGA) rule files. A single CGA rule file can be used to generate many 3D models using feature attribute information stored in the OSM data.

This appendix describes this workflow, using OSM imagery data sources.

Considerations for setting up the CityEngine scene

When you set up your CityEngine scene of the urban model you intend to convert to MVRsimulation's model format and render in VRSG, keep in mind these three considerations:

- If you intend to export your building models in FBX format, you must have CityEngine Advanced, which supports exporting models in FBX format. (CityEngine Basic does not export models in FBX format.). Both CityEngine Basic and Advanced export models in DAE format, which you can import into Presagis Creator for exporting the model in OpenFlight format.
- When you import source data into the CityEngine scene file, you *must* specify the source data's coordinate system as projected UTM WGS1984, and then choose the corresponding UTM zone. Again, this elevation data must be in the UTM WGS84 projection. (VRSG natively supports geocentric WGS1984 but will support UTM WGS 1984 when the –utmModel flag accompanies the model's entry in the vrsg.clt cultural feature file, as described later.)
- Be sure to import your elevation source data or Terrain Tools-generated heightmap into the City Engine scene file, as this data will be used as the heightmap for your models. Once you add the elevation source data or heightmap, select the option "Align shapes to the terrain" to properly align your models on the terrain elevation. Be sure the elevation or heightmap layer is selected as the heightmap when you align your shapes.

Creating a heightmap from compiled terrain to use in CityEngine

In Terrain Tools you can create a raster image from compiled terrain for cases where a heightmap is needed. For example, you might need to use it as an input to CityEngine for building your own set of urban culture models.

Heightmap rasters can be output in TIFF, JPEG, BMP, or any of Esri's supported raster formats. (For a complete list of supported formats, see https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/list-of-supported-raster-and-image-formats.htm.)

Along with the raster, the Create Heightmap tool creates the required accessory files.

To create a heightmap from compiled terrain tiles:

1. In the MVRsimulation Utilities toolset (located in the MVRsimulation Tools toolbox), double-click the Create MDS Heightmap tool.

2. When the Create MDS Heightmap dialog box appears, supply the extents, tile directory, or directories storing the tiles from which to base the heightmap, target location of the heightmap, and a filename for the output heightmap, with its extension. The Extents options are standard Esri options. The example above uses the simplest option for this heightmap purpose, which is to zoom in to the area of interest and choose the Extents option "Same as Display." The other Extents options are listed below:
3. Optionally, specify a Heightmap Pixel Size, which defines the resolution of the raster, in meters per pixel.
4. Click OK.

Exporting the model from CityEngine

When you export your urban model from CityEngine, be sure to set the following:

- For exporting a model to FBX format, choose Advanced Settings > File Type, and then select Text from the drop-down menu to export the model in ASCII format.
- Choose Geometry Settings > Global Offset, and then select Center to center the origin of the model before exporting the model.
- Choose General Settings > Terrain Layers, and then select "Do not export any terrain layers."
- Choose AutoDesk FBX as the output format, so that you can use MVRsimulation's FBX conversion utility to convert the model to MVRsimulation's model format.
- Choose Collada DAE as the output format, if your modeling tool is Presagis Creator (you can choose a Collada version as well). You would import the DAE model in to Presagis Creator to convert it to FLT format, or simply use MVRsimulation's HPX plugin for Creator.

CityEngine does not export levels of detail (LODs) with a model. If you want your resulting model to have LODs when it is rendered in VRSG, you must add them in the appropriate model editing application (such as Autodesk Maya or 3ds Max for FBX models; Presagis Creator for FLT models) prior to converting the model to MVRsimulation's model format.

Converting FBX or FLT model to MVRsimulation's model format

MVRsimulation's Fbx2Hpx command-line utility, which is delivered with VRSG, converts an FBX-formatted model to MVRsimulation's HPX model format for rendering in VRSG. The utility is located in \MVRsimulation\Common\Util\Fbx2Hpx.

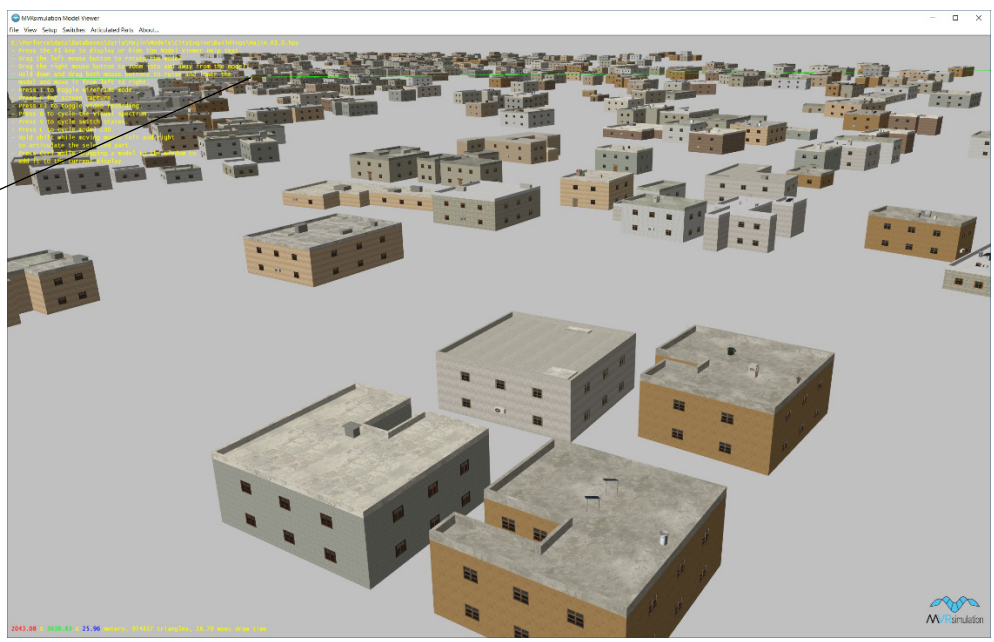
To use the FBX conversion utility, type "Fbx2Hpx" in a command-line window, followed by the name of the model you want to convert, with or without parameters. For more information about using the utility, see the chapter "Converting FBX and OpenFlight Formats to MVRsimulation Runtime Formats" in the *MVRsimulation VRSG User's Guide.*

To convert an OpenFlight format model to an MVRsimulation HPX-formatted model, you can use one of the following methods:

- Within Presagis Creator, use MVRsimulation's hpxPlugin.dll converter. Simply install this plugin in the \Plugins directory of Creator and then restart Creator. The MVRsimulation HPX model format will be available as an export option.
- Use MVRsimulation's oflt2Hpx.exe converter, by typing "oflt2Hpx" in a command-line window, followed by the name of the model you want to convert, with or without parameters.

These utilities are delivered with VRSG and are located in \MVRsimulation\Common\Util\OpenFlight. For more information about using the FLT conversion utilities, see the chapter "Converting FBX and OpenFlight Formats to MVRsimulation Runtime Formats" in the *MVRsimulation VRSG User's Guide.*

After converting the urban model to MVRsimulation's HPX format, you can inspect the model in MVRsimulation's Model Viewer to ensure that the origin of the model is where it is expected and that the model converted properly.



The Model Viewer displays the X (red), Y (green) and Z (blue) axes, which meet at the origin of the model.

For brief information about Model Viewer, see the chapter "Visualizing Terrain." For more complete information, see the *MVRsimulation VRSG User's Guide*.

Visualizing the model on 3D terrain in VRSG

To visualize the converted urban model on 3D terrain in VRSG, you first add an entry for the model to the vrsg.clt cultural feature file used for the terrain. This entry must be added to

vrsg.clt by hand. Although CityEngine-generated models are useful to display on the terrain in Scenario Editor while you are adding other content, these models are too big to place on the terrain directly in Scenario Editor by a drag-drop action. As CityEngine-generated models are tied to a specific footprint and elevation, there is no need to manipulate them in Scenario Editor.

To add an entry to a terrain's vrsg.clt file:

1. Locate the coordinates of the origin of the model.
2. In a text editor, create a new entry for the model in the terrain's vrsg.clt cultural feature file (or an appropriate vrsg.clt file, if your terrain uses more than one). In this entry, add the coordinates of the origin of the model. Add the –utmModel flag to the end of the entry to ensure proper alignment of this UTM-projected model in VRSG.

   For example:

   ```
   N25 12 35.962 E055 16 39.895 6.00 0.00 0.00 0.00
   Dubai_Buildings.hpy -utmModel
   ```
3. Start VRSG.
4. On the Dashboard's Startup Parameters tab, set the search path, ensuring that the vrsg.clt file and model are both in the search path.
5. Launch the VRSG visualization session.
6. Navigate to the area of interest and inspect the model.

Compiling CityEngine models from a vrsg.clt into the terrain

As an alternative to the traditional approach of handling culture models externally in a cultural feature file (vrsg.clt file) for rendering in VRSG at run time, geolocated models can be compiled directly into the terrain. For large areas of CityEngine content, this approach

offers several advantages over loading culture into VRSG via an external vrsg.clt file. The most obvious benefit is that a complied model reduces VRSG startup time. Compiled-in models are serialized in binary format rather than in a .clt's ASCII format, which is much faster to read from disk. Additionally, the compiled-in models leverage VRSG's terrain paging mechanism. VRSG loads the models when the terrain comes into view, not at startup time with the rest of the models listed in the vrsg.clt file. As such, the compiled-in models will not persist in VRSG memory when the tiles are out of eyepoint range. To compile the models listed in a vrsg.clt file into a terrain tile, you simply import the vrsg.clt in terrain Tools and recompile the cultural features for the terrain.

1. In the MVRsimulation Tools Toolbox, expand the MVRsimulation Utilities toolbox and double-click the tool Import Features From CLT File.
2. For the CLT File, click the browse button and find the vrsg.clt that lists the models you want to compile into the terrain.
3. For Output Workspace, choose an output location for the resulting feature class.
4. Click OK to close the dialog box. After the tool finishes executing, a feature layer called "vrsg" will appear at the top of the TOC.
5. Move the "vrsg" layer so that it is listed under Features.
6. Right-click on this new layer, and select Open Attribute Table.
7. Because CityEngine models are not part of the VRSG model libraries, you need to direct Terrain Tools where to look for it. This location can be a full path or a relative path. You could store it in a \User subdirectory in the installed \MVRsimulation\VRSG\Models.

   Alternatively, you can add the path to the model to the Terrain Tools model search path via a modelSearchPath .ini preference. This preference, which you would add to the file \MVRsimulation\ArcGIS Terrain Tools\Config\ TerrainTools_<*user*>_user.ini, is a list of semicolon delimited path entries, each of which Terrain Tools uses to locate models for the active build that are not stored in MVRsimulation's model libraries (\MVRsimulation\VRSG\Models). For example:

   ```
   modelSearchPath = "D:\Perforce\kmolloy-desktop\QA\Terrain
   Tools\Sample_Data\Models;D:\Perforce\kmolloy-desktop\QA\Terrain
   Tools\Customer\kb\spain_bridge"
   ```

8. Open the Terrain Tools Build Manager. On the Build History page, select the build that was used to compile the relevant terrain and click Rebuild Culture, to have the model compiled in with the previously built tile.

APPENDIX D

Using Lidar Data for Building 3D Terrain

LIDAR point clouds can be used in Terrain Tools by converting them to a raster file. This appendix describes a simple workflow for creating a digital elevation model (DEM) raster in ArcMap from point clouds. You can try out these steps with the sample project that is delivered with Terrain Tools. Two Lidar data files (.las files) are located in \MVRsimulation\ArcGIS Terrain Tools\Sample Project\LIDAR\.

MVRsimulation uses this workflow for creating DEMs of small areas (recommended, due to the density of data) as part of creating terrain of airfields and MOUT sites, where the terrain will be used for ground-based simulation.

Overview

Lidar point clouds consist of millions of data points and are stored as a 3-dimensional data cloud as a series of x (longitude), y (latitude), and z (elevation) points. From this dense collection of Lidar point cloud data, you can extract the ground data, convert it to a DEM raster, and then use this raster as an elevation source in Terrain Tools for building 3D terrain.

Creating a LAS dataset

Initially, you need to create a LAS dataset file (.lasd), which points to the LAS files. This dataset file does not import Lidar point data, it only stores the reference information to LAS files. Making the dataset eases the handling of multiple LAS files by working with them as a group.

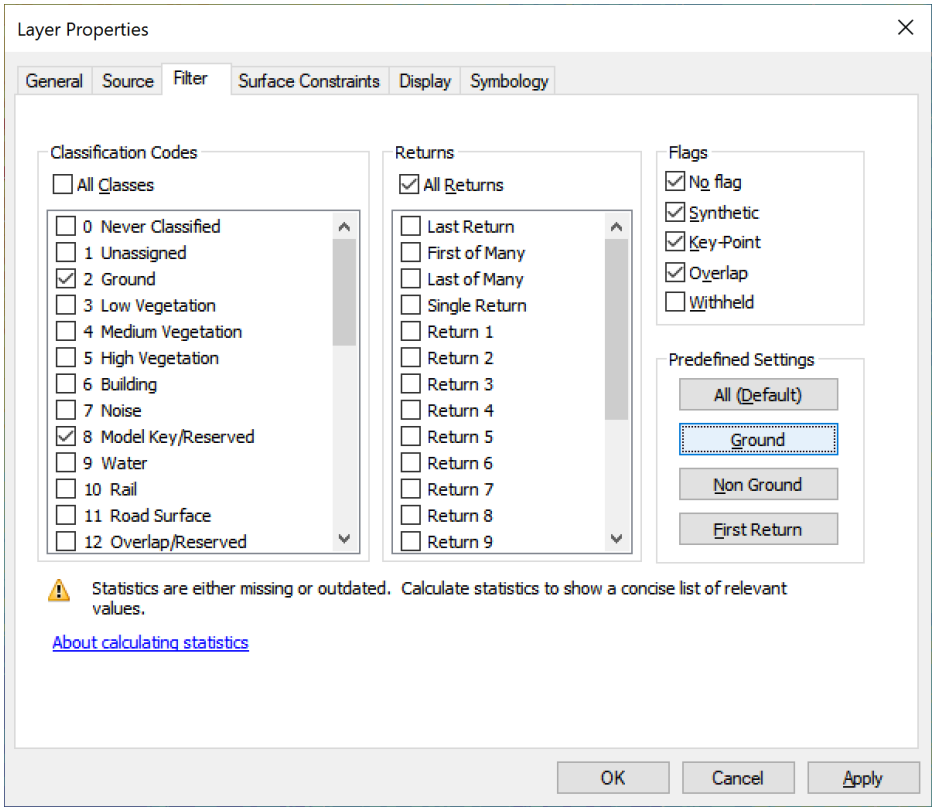
To create a LAS dataset:

1. In ArcMap, open the project for which you want to add Lidar-based DEM, and open the Create LAS Dataset tool, located in the Data Management Tools, LAS Dataset toolbox.
2. When the Create LAS Dataset dialog box appears, browse to the input .las files. For this example, you can use the sample project .las files located in: C:\MVRsimulation\ArcGIS Terrain Tools\Sample Project\LiDAR.
3. Specify the path of the output dataset file (.lasd) and click OK.
4. When the generation of the LAS dataset is complete, it will be added automatically as a layer to the project's TOC.
5. (Optional) To see the point data, right-click on the .lasd dataset layer in the TOC and select Zoom To Layer. If the point data is not visible, you might need to zoom in closer on the map until the point data appears in the map viewer.

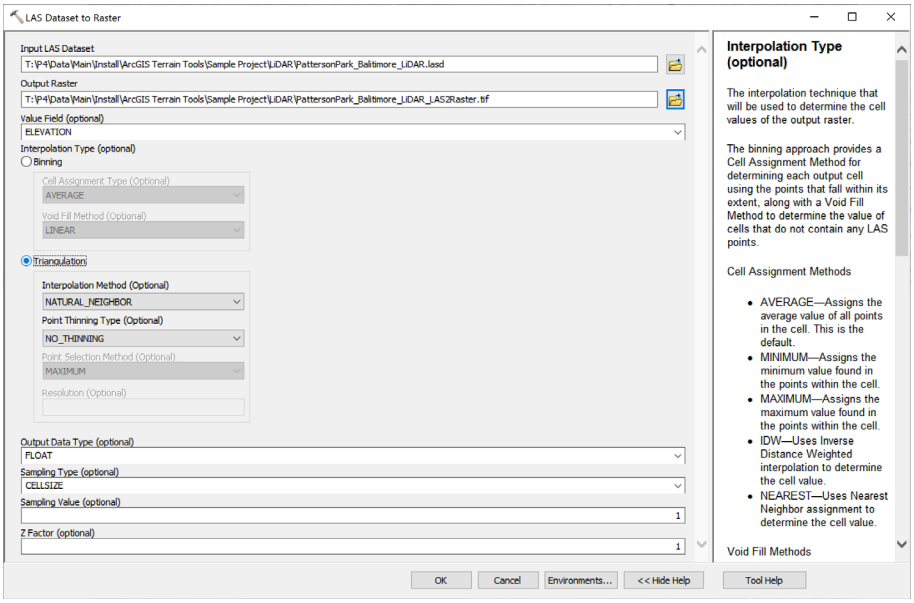
Converting the LAS dataset to a raster

Next you convert the LAS dataset (.lasd) to a raster DEM, which can be used in your terrain project.

1. Locate the .lasd dataset layer in the TOC, right-click on the layer, and from the context menu choose Properties.
2. When the Layer Properties dialog box appears, click the Filter tab, select the Ground Classification Codes checkbox on the left.



3. In the Predefined Settings section on the right, click the Ground button, and then click OK. This action selects only the points classified as Ground in the data, for creating the DEM raster.
4. In the Conversion Tools Toolbox, select the To Raster Toolbox and open the LAS Dataset to Raster tool.
5. When the LAS to Raster dialog box appears, under Input LAS Dataset, select your LAS dataset from the dropdown menu. Optionally, you can drag and drop it into this field.



6. For Output Raster, browse to the intended location and name the output file. Specify the file extension, such as .tif to generate a GeoTIFF, as shown in this example.
7. For Interpolation Type, select Triangulation and then Natural Neighbor.
8. Enter the Sampling value. For this example, use a sample value of 1 to define the resolution of the output.
9. Click OK.

The output raster should look similar to the following image:

10. Remove the .lasd file from the TOC.
11. Place the new raster you just created in the TOC under the Elevation group layer.

Lidar resources

In the US, Lidar data can be obtained from multiple sources, such as the collection of the USGS National Map at www.usgs.gov/core-science-systems/national-geospatial-program/national-map.

Note: LIDAR data file collections are often delivered in a LAZ archive format. Although you cannot uncompress LAZ archive files in ArcMap, you can uncompress them with:

- Laszip, from the suite of open-source LAStools at http://lastools.org/.
- Esri ArcPro.

APPENDIX E

Third-Party Legal Notices

MVRsimulation Terrain Tools for ArcGIS includes open source software licensed under terms that require MVRsimulation to include the following notices.

Twisted

Zope

Zope Public License (ZPL) Version 2.1. A copyright notice accompanies this license document that identifies the copyright holders. This license has been certified as open source. It has also been designated as GPL compatible by the Free Software Foundation (FSF).

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source code must retain the accompanying copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the accompanying copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Names of the copyright holders must not be used to endorse or promote products derived from this software without prior written permission from the copyright holders.
4. The right to distribute this software or to use it for any purpose does not give you the right to use Servicemarks (sm) or Trademarks (tm) of the copyright holders. Use of them is covered by separate agreement with the copyright holders.
5. If any files are modified, you must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

Disclaimer:

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Jinja2

Chromium Embedded Framework

APPENDIX F

License and Warranty

MVRsimulation Inc. Software License Agreement

Version 44930 (5 February 2023)

DO NOT INSTALL THIS SOFTWARE UNLESS YOU HAVE READ, UNDERSTOOD AND AGREE TO THESE TERMS

MVRsimulation Inc. ("**MVR**"), a Massachusetts corporation, is supplying a copy of its VRSG®, Scenario Editor or Terrain Tools software to You (as that term is defined below) on the terms set forth herein and in the Purchase Order (as defined below). This License Agreement is entered into as of the date first written above between MVRsimulation and the undersigned licensee ("**you**"), with an address indicated below your signature line. Now, therefore, you and MVRsimulation agree as follows:

Article 1 **DEFINITIONS:** As used herein:

"**Additional Materials**" means software, libraries and databases other than VRSG, Scenario Editor or Terrain Tools or Updates to either VRSG, Scenario Editor or Terrain Tools, including, without limitation, MDS Read API, MDX Read API and the MVR Viewpoint Protocol.

"**Authentication Files**" means the files in .C2V, .V2C or .A2C format that are exchanged in the course of a License Authentication or the reinstallation of the Software onto another computer.

"**Data Files**" means data files created or supplied by MVR under a Purchase Order or as Additional Materials or created by you or MVR as part of the License Authentication process.

"**Demonstration Video**" includes any visual display involving VRSG, Scenario Editor or Terrain Tools or the Data Files that can be viewed by a third party, whether hosted on a public server, circulated in a format that permits individual viewing, or demonstrated at your facility or that of a third party.

"**Documentation**" means documentation published by MVR describing the functions and operation of the Software.

"**Dongle ID**" means a unique digital identifier of a physical dongle transmitted as part of the authentication process described in Section 2.02.

"**Initial Maintenance Term**" is a one year term beginning on the date on which you first receive a long-term License Authentication or, if different, the maintenance term is specified in the Purchase Order.

"**License Authentication**" means the authentication described in section 2.02 below.

"**License ID**" means a unique digital identifier of a physical computer installation transmitted as part of the authentication process described in Section 2.02.

"Maintenance Term" means the Initial Maintenance Term as it may be extended from time to time pursuant to Section 3.04.

"Materials" means the Software and related Documentation.

"**Proprietary Data Files**" means Data Files provided in any proprietary data format of MVR, including without limitation data provided in the .VIR, .MDY, .MDX, .MDS, .HPZ, .HPY, .HPX, .TEX, .C2V, .V2C and/or .A2C format.

"**Purchase Order**" means the purchase order, quotation or online order form, including the Standard Terms, pursuant to which you have agreed to purchase and MVR has agreed to sell licenses of Software.

"**Refresh**": A License Authentication may be "**Refreshed"** if you transfer your Software to a new computer or request the right to use additional Software features beyond those in your original Purchase Order. This refreshing may take the form of reprogramming of a dongle or the exchange of new Authentication Files.

"**Scenario Editor**" means MVRsimulation's VRSG Scenario Editor software product.

"**Software**" means the MVR software (which may include VRSG) identified in the Purchase Order and any entity libraries, Data Files, upgrades, updates, new releases, versions, corrections or revisions thereto which MVR makes or has previously made available to you.

"**Standard Terms**" means MVR's standard terms and conditions available at www.mvrsimulation.com/howtobuy/standard_terms_and_conditions.html or, if attached as an exhibit to the Purchase Order or this license agreement, in the form so attached.

"**Terrain Tools**" means MVR's Terrain Tools for Esri ArcGIS software product.

"**Updates**" means updates, upgrades and new releases and subreleases of VRSG, Scenario Editor or Terrain Tools software product.

"**VRSG**" means MVR's Virtual Reality Scene Generator software product.

"**You**" means the legal entity or governmental agency on behalf of which the user of this copy of VRSG, Scenario Editor or Terrain Tools is acting in installing and using this copy of VRSG, Scenario Editor or Terrain Tools. This definition applies even when "you" appears with a lower case 'y'.

Article 2 LICENSE

Section 2.01 **Grant**. MVR grants you a non-exclusive license (i) to install the Materials onto a computer under your control, whether the computer is accessed directly or over a private network and when installed over a private network, and (ii) for you or your employee (and, if you are an accredited college or university, your students) to use the Data Files and Software so installed for your own internal business, operational or educational purposes, provided, however that the Data Files and Software may only be run or otherwise used on a computer that has been authenticated, as described in Section 2.02.

Section 2.02 **Authentication**. Authentication of a computer may occur either as a result of an exchange of Authentication Files or by your use of a dongle obtained from MVR. If your Purchase Order refers to authentication by means of Authentication Files, the exchange of Authentication Files will occur when you install the Software on a computer. As part of the installation process, the

Software will collect and compile unique identity information of the physical computer either (i) into a .C2V file, or (ii) into a License ID output to the computer's video display, to be transmitted via email to MVR. Upon receipt of the .C2V file or the License ID code, MVR will create and send by reply e-mail either a .V2C or .A2C Authentication File which you will be able to use to complete the authentication process. MVR's transmission to you of the Authentication File will trigger the commencement of the Initial Maintenance Term for the copy of VRSG, Scenario Editor or Terrain Tools to which it is related.

For users whose authentication is to be done by means of dongle(s), MVR will provide the number of dongles stated on the Purchase Order. An individual dongle may not be used by more than a single user or on more than a single computer simultaneously. As part of the installation process, you will be prompted to send the applicable Dongle ID(s) via email to MVR. Upon receipt of these Dongle ID(s), MVR will send by reply email an unlock code for each dongle for which a Dongle ID has been received. You will be able to use these unlock code(s) to complete the authentication process. MVR's transmission to you of the unlock code for a dongle will trigger the commencement of the Initial Maintenance Term for the copy of VRSG, Scenario Editor or Terrain Tools to which it is related.

Although MVR may provide you with temporary License Authentication (which may involve a temporary unlock code for a dongle or a .A2C or .V2C authentication file with a temporary duration), MVR shall have no obligation to extend any such temporary License Authentication, provide additional temporary License Authentication or provide you with a permanent License Authentication unless and until you have paid in full for the Software and any other services or materials identified in the Purchase Order. Accordingly, your use of the Software may be interrupted until such payment has actually been received and processed by MVR.

Section 2.03 **Restrictions**. You may not do or permit any other party to do any of the following:

a) Use the Materials other than as specifically permitted in this Agreement;
b) Operate the Software for use by, or make the Data Files available to, persons other than those described in clause (ii) of Section 2.01 or more concurrent users than the number of licenses that you have purchased;
c) Permit simultaneous use of the Materials by more users than are authorized by the License Authentication, whether over a computer network, or on a virtual or emulated computer or otherwise;
d) Attempt to (a) alter, merge, modify, adapt, or translate the Software (including without limitation any Proprietary Data Files but excluding Data Files that are not Proprietary Data Files), (b) decompile, reverse engineer, disassemble, derive, or otherwise reduce the Software to a human-perceivable form, or (c) develop any software that would permit an end user to read or access the Proprietary Data Files. The parties understand that, as used herein, the term "reverse engineer" shall include, without limitation, any use of benchmarking information or incremental output from the Software to determine MVR source code, algorithms or data format, or for the purpose of recreating the Software (including without limitation any MVR Files) or creating software or files substantially similar thereto;
e) Bypass the copy protection code or any other technological measure that controls access to the Software or Materials;
f) Make copies of the Materials other than one copy for back-up or archival purposes or use a back-up copy other than as a replacement for the original copy. You must include on any back-up copy all copyright and other notices included on the Materials;
g) Export, re-export or use the Materials or any copy thereof in violation of the export control laws of the United States of America or any other country;

h) Use any dongle or Authentication File supplied to you by MVR in connection with the license of the Software in any manner other than in connection with the use of the Materials as permitted hereunder;
i) Use the Software to test or analyze the performance or user interface of the Software in order to develop or improve a product which competes with the Software;
j) Publish or provide to third parties performance characteristics relating to the Materials without the express written consent of MVR;
k) Demonstrate the Materials in public or private forums without using (i) a platform that provides the Software sufficient performance capacity to operate at peak capacity, and (ii) the most recent versions of any Software and Data Files;
l) Use or circulate any Demonstration Video without first obtaining the approval of MVR, which approval MVR may withhold if it reasonably determines that the Demonstration Video will not correctly reflect the peak performance of the Software and Data Files;
m) Publish or provide this document to third parties in electronic or printed form;
n) Create or distribute derivative works based upon the Software or Materials (including without limitation databases derived from files provided to you by MVR).

Section 2.04 **Updates to License Agreement**. If you renew, revive or extend the Maintenance Term, Refresh your License Authentication, or request and receive any Updates or Additional Material, you may be prompted to accept the then-current version of this Agreement, which if accepted shall apply to all Materials licensed hereunder. You will not be able to use Updates or Additional Material or extend or revive the Maintenance Term without accepting the then-current version of this Agreement unless MVR then expressly permits it (which it may refuse in its absolute discretion), but any extension of the Maintenance Term previously paid for will continue in effect under the existing Agreement until it expires.

Section 2.05 **Ownership and Copyright**. Title and copyright to the Materials (including, without limitation, any databases, libraries, images, "applets", photographs, animations, video, audio or music and text incorporated therein and any hardware keys provided in connection therewith) and all copies thereof remain with MVRsimulation and/or its licensors. The Materials are copyrighted and are protected by United States copyright laws and international treaty provisions. You may not remove the copyright and other proprietary rights notices from the Materials. You agree to prevent any unauthorized copying of the Materials. Except as expressly provided herein, MVR does not grant any express or implied right to you in the Materials or under the patents, copyrights, trademarks, or trade secret information of MVR or its licensors.

Article 3 MAINTENANCE AND SUPPORT

Section 3.01 **Initial Support**. Except as you and MVR may otherwise have mutually agreed in writing, MVR will provide technical support and maintenance in connection with your use of the Materials during the Initial Maintenance Term. If MVR later provides you with any subsequent License Authentication (as may arise if the Software is transferred to a replacement computer, or MVR provides you with a replacement dongle or any new unlock code for a dongle), your subsequent License Authentication shall not restart or extend the Maintenance Term. If you have purchased multiple seats of the Software and have therefore received multiple License Authentications, the Materials accessed with each License Authentication will have a separate Maintenance Term, based on the date on which MVR first provides you with a long-term License Authentication for the applicable copy of the Software. Upon the expiration or termination of the Maintenance Term, MVR will have no obligation to provide maintenance or support for the Materials. MVR may temporarily suspended its support in the event of interruptions beyond its reasonable control, such as may arise in

the event of flood, earthquake, terrorist attack, failure of third party communications systems and the like.

Section 3.02 **Nature of Support**. All maintenance and support will be provided remotely during MVR's normal business hours. The maintenance and support provided by MVR hereunder will consist of:

a) responding promptly via e-mail during the Maintenance Term to any questions regarding reports of errors or defects in the Software;

b) providing assistance via e-mail and telephonically during the Maintenance Term relating to the installation and use of the software;

c) if you first received the Materials electronically, permitting you to re-install the Materials electronically on the same or a different computer until the end of the Maintenance Term or, if longer, until the fifth anniversary of the end of the Initial Maintenance Term. However, if you extend the Maintenance Term beyond six years, MVR reserves the right to provide you with a later version of the Materials than the version that was specified in your Purchase Order; and

d) providing access during the Maintenance Term to (1) any "Additional Materials" generally released to MVR's customers, (2) error corrections (i.e. patches) and updates intended to fix reported errors, and (3) all product updates, upgrades and enhancements to the Software that MVR generally releases to its customers during the Maintenance Term.

If you request that MVR perform maintenance or support on site at your premises or outside normal business hours, you will be responsible for all reasonable expenses incurred by MVR in connection with performance thereof and MVR reserves the right to charge its then prevailing rates per hour of service provided, or to refuse to provide such service.

Section 3.03 **Exclusions and Limitations**. MVR's maintenance and support obligations do not include: custom programming, training, implementation, database changes or support, product upgrades for which MVR generally imposes a separate price or fee, any requests for content revisions or changes of any kind, or any other matters not specifically covered hereunder. In addition, the provision of maintenance or support hereunder is subject to commercial availability and technological compatibility and the absence of any actual or threatened litigation between you and MVR. MVR provides no guarantee that the Software or any upgrades or updates to the Software provided pursuant to this Maintenance Agreement will function or perform when used on equipment that does not conform to the specifications indicated in the documentation relating thereto. MVR shall have no obligation to provide maintenance or support if you use or attempt to use the Software with hardware that fails to meet the minimum requirements indicated in the Documentation or otherwise modify, revise or transfer the software other than as permitted herein or without MVR's prior written permission.

Section 3.04 **Extension of Maintenance Term**. At any time before the expiration of the Maintenance Term associated with a particular copy of the Software, you may extend such Maintenance Term for successive one (1) year terms, provided, however, that such extensions shall no longer be available or applicable if MVR has generally discontinued maintenance of the licensed version of the Software (in which case MVR shall refund any amounts that you have already paid for extensions that have not yet commenced). For each annual extension of the Maintenance Term associated with a copy of the Software, you shall pay MVR the applicable maintenance fee set forth in the Purchase Order for extension of the Maintenance Term or, if an applicable maintenance fee is not

specified, the price then generally charged by MVR for the maintenance and support services described herein. All payments for extensions purchased hereunder must be received by MVR before the expiration of the then current Maintenance Term.

Section 3.05 **Termination of Maintenance Term.** The Maintenance Term will terminate automatically in the event that you (a) breach any term of this Agreement (including without limitation any payment obligation contained herein), or (b) violate or infringe any of MVR's intellectual property in any manner unless such proposed termination is governed by the Contract Disputes Act of 1978, in which case MVR may pursue its rights in the manner prescribed in that Act and the regulations promulgated thereunder.

Section 3.06 **Effect of Termination or Expiration**. The Software and any updates, upgrades and enhancements thereto that are installed during the Maintenance Term will continue to be accessible after the expiration or termination of the Maintenance Term. However, even though you may be able to download and install product updates, upgrades and enhancements to the Software after the expiration or termination of the Maintenance Term, these updates, upgrades and enhancements will not function on your computer unless and until you have revived and reinstated the Maintenance Term as described below.

Section 3.07 **Revival**. In the event that you wish to receive maintenance and support after the expiration or termination of the Maintenance Term, you may request that the Maintenance Term be revived and reinstated for a new one (1) year term. MVR may, accept or refuse such a request in its sole and absolute discretion. Upon notice that MVR has agreed to revive the Maintenance Term, you shall pay to MVR a maintenance fee at its then-applicable rates for (a) the one (1) year term commencing as of the date when MVR receives the payment and (b) any gap period (measured in months) between the end of the expired or terminated Maintenance Term and the commencement of the newly revived one-year term described in clause (a).

Article 4 SAFEKEEPING OF DONGLES AND AUTHENTICATION FILES

If you have been provided a dongle or an Authentication File format in connection with your license of Software, the safekeeping of the dongle or Authentication Files is your responsibility. MVR has no responsibility to replace a dongle or Authentication Files that has been lost. **For that reason, MVRsimulation strongly recommends that you back up your .A2C or .V2C files to a separate storage device and retain that backup file for archival purposes.** MVR will however replace a damaged dongle if you physically return it (including the casing with identifiable labeling and all mechanical and electrical parts) to MVR in a manner sufficient to allow MVR to troubleshoot any damage thereto and confirm to MVR's satisfaction that there has been no attempt to use the dongle in a manner not permitted under this Agreement.

Article 5 LIMITED WARRANTIES; DISCLAIMERS; REMEDIES; INDEMNITY

Section 5.01 **Limited Warranties**. MVRsimulation warrants that:

(a) The Software will perform substantially in accordance with the technical functionality set forth in the Documentation during the Maintenance Term; and

(b) Any physical media on which the Materials are delivered will be free from defects in material and workmanship that will prevent you from loading the Software on your computer for a period of sixty (60) days from the date of shipment to you.

These warranties shall be null and void in the case of any defect caused by any of the following:
(i) modification of the Materials by any party other than MVR;

(ii) use of the Materials with hardware or software other than that supplied or recommended by MVR;

(iii) other improper or unauthorized use of the Materials by you; (iv) failures or defects in third party software or hardware; or (v) external factors such as, but not limited to, power failures or electrical surges.

Section 5.02 **Remedies**. If the Software fails to perform substantially in accordance with the Documentation, your sole remedy is to initiate a technical support ticket by contacting MVR at support@mvrsimulation.com; and MVR's sole obligation will be to provide reasonable commercial efforts to resolve the issue to your satisfaction in accordance with Section 3.02. Your sole and exclusive remedy with respect to any defective media shall be the right to return such media to MVR, and MVR's sole liability to you shall be the replacement of any defective media.

Section 5.03 **DISCLAIMERS.** EXCEPT AS SET FORTH ABOVE, THE MATERIALS ARE PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF SATISFACTORY QUALITY, MERCHANTABILITY, NONINFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, NO WARRANTY IS MADE THAT THE ENCLOSED SOFTWARE WILL GENERATE COMPUTER PROGRAMS WITH THE CHARACTERISTICS OR SPECIFICATIONS DESIRED BY YOU OR THAT THE MATERIALS WILL BE ERROR-FREE. THE WARRANTY PROVIDED HEREIN DOES NOT EXTEND TO ANY HARDWARE PURCHASED FROM MVRSIMULATION. ANY SUCH WARRANTY MUST BE PROVIDED IN A SEPARATE WRITING. THESE DISCLAIMERS OF WARRANTY CONSTITUTE AN ESSENTIAL PART OF THIS AGREEMENT.

Without limiting the foregoing, MVR provides no guarantees that the Software or any upgrades or updates to the Software provided as part of the maintenance and support described below will function or perform when used on equipment that does not conform to the specifications indicated in the Documentation relating thereto.

Because certain jurisdictions prohibit the waiver of certain warranties, the above disclaimer may not apply to you and you may have additional legal rights that vary by jurisdiction.

Section 5.04 **LIMITATION ON LIABILITY**. TO THE MAXIMUM EXTENT PERMITTED BY LAW, IN NO EVENT SHALL MVRSIMULATION OR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION OF THE MATERIALS BE LIABLE FOR ANY DAMAGES OR LOSSES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DIRECT OR INDIRECT DAMAGES, INCIDENTAL OR CONSEQUENTIAL DAMAGES, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION AND/OR LOSS OF DATA), WHETHER ARISING IN TORT, CONTRACT OR OTHERWISE, ARISING OUT OF THE USE OF OR INABILITY TO USE THE MATERIALS, EVEN IF MVRSIMULATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, MVRSIMULATION'S ENTIRE LIABILITY HEREUNDER SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID FOR THE MATERIALS.

Because some jurisdictions prohibit the exclusion or limitation of liability for damages, the above limitation may not apply to you and you may have other legal rights that vary by jurisdiction.

Section 5.05 **Indemnity of MVRsimulation**.
a) Commercial Customers. If you are not an agency of the United States Government, you agree to indemnify MVR and its affiliates against any loss, liability or expense (including reasonable legal

fees) it incurs arising out of or in connection with any breach or violation of the terms of this Agreement by you or your use of the Materials.

b) Purchases By U.S. Government. If you are the United States government or an agency thereof, then you authorize and consent to all use and manufacture, in performing this license, of any invention described in and covered by a United States patent,

(1) Embodied in the structure or composition of any article the delivery of which is accepted by the Government under this contract; or,

(2) Used in machinery, tools, or methods whose use necessarily results from compliance by MVR with (i) specifications or written provisions forming a part of this contract or (ii) specific written instructions given by the Contracting Officer directing the manner of performance. The entire liability to the Government for infringement of a United States patent shall be determined solely by the provisions of the indemnity clause, if any, included in this license, and you assume liability for all other infringement to the extent of the authorization and consent hereinabove granted.

Article 6 CONFIDENTIALITY

You agree that the Materials, the object and source code of the Software, the algorithms used by the Software, the performance characteristics of the Software, and the algorithms and functioning of any dongles or Authentication Files provided to you in connection therewith (collectively, the **"Confidential Information"**) are or contain the confidential and proprietary information and trade secrets of MVR and that MVR is providing the Materials to you in confidence. You shall not and you shall not permit others to reverse engineer the Software (including by analysis of benchmarking or output information) or to access the source code, algorithms or performance characteristics of the Software, the dongles or the Authentication Files. In addition, you agree (i) to preserve in strictest confidence all Confidential Information, (ii) not to disclose the Confidential Information to any third party except as expressly permitted herein, (iii) only to disclose the Confidential Information within your business organization to those employees (and, if you are an accredited college or university, your students) who have first agreed to be bound by the terms and conditions substantially similar to those contained herein, (iv) not to disclose any Confidential Information to any agents, contractors or consultants except if such disclosure is related to the authorized use of the Materials hereunder and after having received a commitment from such agents, contractors or consultants to be bound by substantially similar obligations with respect to such information as you are hereunder, and (v) not to use the Confidential Information for any reason except in connection with the authorized use of the Materials. You shall be responsible and liable for any unauthorized disclosure, publication or dissemination by any of your employees, students, agents or contractors of any Confidential Information. Confidential Information shall not include any information which: (a) you possess prior to the receipt hereof without obligation of confidentiality; (b) you rightfully receive from a third party without any obligation of confidentiality to such third party, and which such third party received without any obligation of confidentiality, direct or indirect, to MVR; or (c) is or becomes publicly available lawfully and without breach of any obligation to MVR by you. You may disclose Confidential Information if such disclosure is required under the terms of any statute, regulation, order, subpoena or document discovery request, provided that prior written notice of such disclosure is furnished to MVR as soon as practicable in order to afford MVR an opportunity to seek a protective order or otherwise contest or restrict such required disclosure. The parties agree to cooperate fully to limit disclosure in the event of any apparent legal requirement that Confidential Information be disclosed.

Article 7 TRANSFER AND ASSIGNMENT

Section 7.01 **Transfer or Assignment by Licensee**. Except as otherwise permitted by law or expressly permitted herein, you may not transfer or assign this Agreement or the Materials to another person without the prior written permission of MVR, except:

(i) if the Purchase Order identifies a U.S. Department of Defense ("DOD") contract in furtherance of which you have ordered the Materials, this license may be transferred to any other DOD contractor who needs the Materials in furtherance of a DOD contract or to the United States government agency for which the contract is being performed,

(ii) in connection with the sale of all or substantially all of your assets, this license may be transferred to the purchaser, and

(iii) if you are acting as a systems integrator for an end user identified in the Purchase Order, this license may be transferred to the end user;

provided, however, that (x) you are then in compliance with your payment obligations under any related Maintenance Agreement then in effect with respect to the Materials, and (y) the transferee provides MVR with:

(A) an unqualified, written acceptance of the terms of this License and any related maintenance agreement fifteen (15) days after the transfer (and, for transfers pursuant to clause (i), identifying the DOD contract for which the Materials are required), and

(B) the name, address, telephone number and e-mail address of an employee of the transferee authorized to communicate with MVR in connection with this License and any related maintenance agreement.

In no case shall the Software and Materials or any related dongle or Authentication File be knowingly or intentionally licensed, transferred or assigned to terrorist sponsored organizations or to organizations which primarily reside within terrorist countries as defined by the United States of America Department of State.

Any transfer made pursuant to this Section must include all of the Software and Materials and any related dongle and Authentication Files. You shall be solely responsible for any transfer being in compliance with United States export laws and regulations. Upon a transfer in compliance with this Section, the transferee shall thereafter be solely responsible for compliance with the terms of this license agreement (except for any breach or violation which predates the transfer, for which you shall remain responsible) and you will have no further obligation to indemnify MVR hereunder except with respect to your use of the Materials prior to the transfer. If you are acting as a systems integrator for an end user, you may only use the Materials to develop, install and support the systems for the end user and not for any other purpose.

Section 7.02 **Transfer to Another Computer**. If you wish to transfer the Software from one of your computers to another, the process involved will depend upon your method of authentication. If you have a hardware dongle, you may install that dongle in the new computer and install the Software on it. If you authenticated by means of Authentication Files, you may (i) transfer the hard drive containing the Software from your old computer to your new one, or (ii) reinstall the Software on a new hard drive by requesting permission to effectuate such a transfer in an e-mail to MVR, which may impose a fee for permitting such a transfer. If MVR permits such a transfer, it will instruct you as to how to generate a License ID or .C2V file that will de-install the Software from the original computer. You must then send a License ID or .C2V cancellation receipt that will be generated as part of the de-installation process to MVR by e-mail, and MVR shall (upon your payment of any fee required as part

of the transfer process) generate a new .A2C or .V2C Authentication File for your use in installing the Software on the replacement computer. **If you are unable to generate a new License ID or .C2V file (which may happen if, for example, your original computer has become lost, stolen or inoperable) then you may be unable to reinstall the software on a new computer other than by using the old hard drive on the new computer**.

Section 7.03 **Transfer by MVRsimulation.** MVR may not assign its rights and delegate its duties hereunder without your prior consent, which you may not unreasonably withhold, condition or delay, except that MVR may freely assign its rights and delegate its duties hereunder to a purchase of substantially all of its assets or to a merger partner in a merger in which MVR is not the surviving entity. After such an event, MVR shall have no further obligation to you hereunder except to provide you, or to cause its successor to provide you, with notice of the transaction within 30 days of its occurrence.

Article 8 SUPPLEMENTAL LICENSE TERMS

Section 8.01 **Additional Materials**. In certain cases MVR may provide Additional Materials at no extra charge to customers with whom MVR has an active maintenance relationship, as part of its maintenance and support. Such Additional Materials are provided under a license that is revocable by MVR without cause in its sole and absolute discretion. Upon request by MVR, you agree, within five (5) days of receiving such request, to return to MVR all of your copies of the Additional Materials, destroy all electronic copies of the Additional Materials in your possession or control, and take such additional actions as MVR may reasonably request to ensure that no copies of the Additional Materials remain in your possession and control. Except that the license to such Additional Materials is terminable by MVR at will, and except as may be otherwise agreed in writing by MVR, the Additional Materials and the use thereof are (i) subject to all of the provisions and restrictions contained in this license as "Software" and (ii) provided solely for the purpose of integration with systems that use VRSG, Scenario Editor or Terrain Tools. MVR makes no representation or warranty regarding Additional Materials and you are responsible for compliance with any export laws and regulations applicable to any export or deemed export of them.

Section 8.02 **Third Party Materials**. MVR may include among the Materials or Additional Materials software, libraries or databases provided by third parties ("**Third Party Materials**"). Although MVR makes these Third Party Materials available for your convenience, in certain cases you will not be able to use or access specific Third Party Materials with, or as part of, the Software until you have first accepted specific terms and conditions provided by the owner of such Third Party Materials (e.g., by executing a clickwrap or license agreement). Your use of any Third Party Materials provided by MVR will be subject to both the terms of this Agreement and any terms and conditions provided by the owner of such Third Party Materials.

Section 8.03 **U.S. Government Restricted And Limited Rights**. The Materials have been developed entirely at private expense and have been sold and offered for sale to non-governmental customers. The Software is "commercial computer software" as defined in DFARS 252.227-7014 (Feb. 2012) and in FAR 2.101(a), and "restricted computer software" as defined in FAR 27-401 (Oct. 2014) (or any equivalent agency regulation or contract clause). The Materials comprising computer software are provided with the rights set forth in FAR 52.227-19 (November 2007). The Materials comprising computer software are provided with the rights set forth in FAR 52.227-19 (November 2007). The Materials comprising technical data are pre-existing technical data developed entirely at private expense and, are provided with the rights described in DFARS 252.227-7015(b) (Feb. 2014). The foregoing grants of Restricted and Limited Rights are only for the benefit of the United States government and its contractors and, in their hands, override any inconsistent restrictions

set forth elsewhere in this License Agreement. The Materials may only be sold or transferred to an agency or instrumentality of the United States Government under prime contracts that effectively incorporate restrictions on government use, reproduction, or disclosure no less protective of MVR than the foregoing and any other attempted sale is null and void. Use, reproduction, or disclosure of the Materials by the government or its agents or contractors is subject to the restrictions set forth herein and/or therein, as applicable. Contractor and manufacturer are MVRsimulation Inc., 57 Union Avenue, Sudbury, MA 01776. Use of the Materials by the United States government constitutes acknowledgment of MVR's proprietary rights in them.

Article 9 TERMINATION

Upon any material violation of any of the provisions of this Agreement, your right to use the Materials shall automatically terminate without reimbursement and you shall be obligated, within thirty (30) days of receiving a notice of termination of this license from MVR, to return to MVR all of your copies of the Materials and any hardware keys provided to you in connection therewith, destroy all electronic copies of the Materials and Authentication Files in your possession or control, and take such additional actions as MVR may reasonably request to ensure that no copies of the Materials or Authentication Files remain in your possession and control. However, the foregoing shall not apply if the matter constitutes a dispute governed by the Contract Disputes Act of 1978, in which case MVR may pursue its rights in the manner prescribed in that Act and the regulations promulgated thereunder.

Article 10 GENERAL

Section 10.01 **Complete Agreement.** This Agreement, the manually signed license agreement between you and MVR (if any) and the Purchase Order constitute the entire agreement between you and MVR and supersedes all representations, understandings and other agreements between the parties with respect to the subject matter described herein or therein. In the event of an express inconsistency between this Agreement or the manually signed license agreement and the Purchase Order, the inconsistency shall be resolved by giving precedence to the inconsistent terms as follows:

(a) first, to any negotiated rider or addendum to a manually signed version of this Agreement (or any proper termination thereof), regardless of whether it is signed before or after the electronic acceptance of this Agreement (with such documents taking precedence with respect to each other in reverse chronological order of their effective dates);

(b) second, to terms specifically added to the Purchase Order as a result of negotiations between the parties;

(c) third, to the terms of this Agreement; and

(d) fourth to preprinted or standard terms of the Purchase Order that were not modified or included as a result of negotiations between the parties.

To establish that Purchase Order terms were negotiated, a party must produce e-mail or other written correspondence pre-dating the execution of the Purchase Order constituting or acknowledging such negotiations. If this agreement is presented in connection with your installation of an Update of VRSG, Scenario Editor or Terrain Tools, it shall supersede any prior electronic version of the license agreement that You accepted upon an earlier installation of this copy of VRSG, Scenario Editor or Terrain Tools or an Update to it but shall remain subject to the documents identified in clauses (a) and (b) above.

Section 10.02 **Amendment and Waiver.** Failure of a party to enforce any provision of this Agreement does not constitute and should not be construed as a waiver of such provision or the right

to enforce such provision. This Agreement may be amended only by a writing executed by both parties or by your electronic acceptance of a more recent version of this license agreement provided to you by MVR.

Section 10.03 **Trademarks.** Nothing contained herein shall give you the right to use any of MVR's trademarks or trade names and you agree not to remove or alter any trademark, trade name, copyright or other proprietary notices, legends, symbols or labels appearing on or in any copies of the Materials.

Section 10.04 **Governing Law; Venue.** This Agreement is governed by the laws of the United States of America and the Commonwealth of Massachusetts, without giving effect to conflict of laws provisions thereof. Any action or proceeding brought by either party against the other arising out of or related to this Agreement shall be brought only (i) in a Massachusetts state court or federal district court for the District of Massachusetts, or, (ii) in the case of a proceeding brought by or against the United States government, the Federal Court of Claims or any successor thereto, and each of MVR and you hereby consent to the personal jurisdiction of such courts. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

Section 10.05 **Customer Suggestions.** MVR considers developing product features requested by customers at no additional cost. MVR retains all rights to these features and may incorporate them in its commercial off-the-shelf products. For further information about MVR's practices in this regard, go to www.mvrsimulation.com/howtobuy/customerfeatures.html.

Section 10.06 **No Corrupt Practices.** You warrant that, in the course of obtaining this license of the Materials, or of selling any products (the "Purchaser Products") into which the Materials are to be integrated, (a) neither you nor your employees or agents have made, offered or promised to make or offer, any payment or any proffer of anything of value, including bribes, either directly or indirectly to any public official, regulatory authority or anyone else for the purpose of influencing, inducing or rewarding any act, omission or decision involving sales of the Materials or the Purchaser Products in order to gain an improper advantage, nor have you authorized or encouraged any other party to do so, and (b) you shall comply, and shall cause your employees and agents to comply, with all applicable anti-corruption and anti-bribery laws and regulations in the course of obtaining purchase orders, requisitions or other authorizations to purchase the Materials and the Purchaser Products. You shall notify MVR immediately upon becoming aware of any breach of your obligations under this Section.

To initiate a support ticket, please contact MVR at support@mvrsimulation.com.

MVRsimulation and the MVRsimulation logo and VRSG (Virtual Reality Scene Generator) are registered trademarks and the phrase "geospecific simulation with game quality graphics" is a trademark of MVRsimulation Inc.

MVRsimulation's round-earth VRSG terrain architecture is protected by U.S. Patent 7,425,952.
All other brands and names are property of their respective owners.