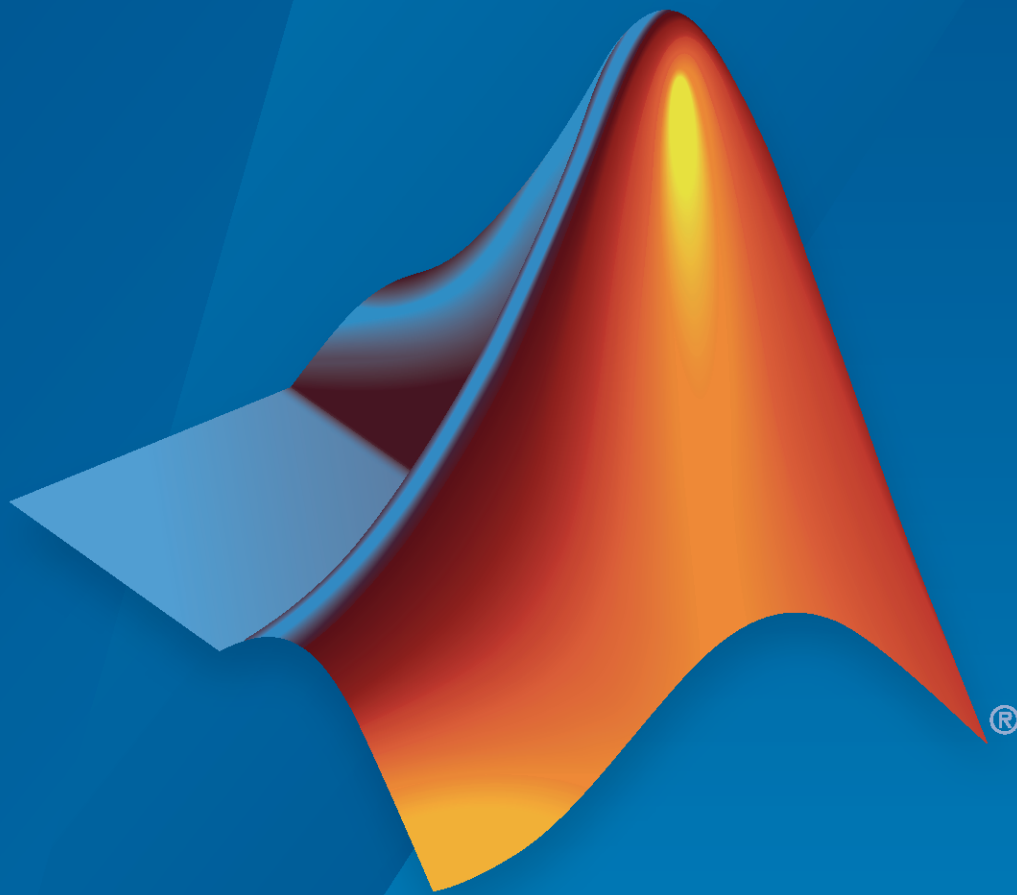


RoadRunner

User's Guide



R2021a

How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

RoadRunner User's Guide

© COPYRIGHT 2020-2021 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

| | | |
|----------------|-------------|----------------------------------|
| April 2020 | Online only | New for Version 1.0 (R2020a) |
| September 2020 | Online only | Revised for Version 1.1 (R2020b) |
| March 2021 | Online only | Revised for Version 1.2 (R2021a) |

1

Get Started with RoadRunner

| | |
|--|-------------|
| RoadRunner Product Description | 1-2 |
| RoadRunner System Requirements | 1-3 |
| Install and Activate RoadRunner | 1-4 |
| Verify System Requirements | 1-4 |
| Get License and Product Installer | 1-4 |
| Install RoadRunner | 1-5 |
| Activate License | 1-6 |
| Create a New Project and Scene | 1-6 |
| Get RoadRunner Updates and Upgrades | 1-8 |
| Update Installed Release | 1-8 |
| Upgrade RoadRunner Release | 1-8 |
| Install RoadRunner Add-On Products | 1-8 |
| Update RoadRunner Licenses | 1-9 |
| Update RoadRunner Individual License | 1-9 |
| Update RoadRunner Network Licenses | 1-9 |
| Install Network License Manager for RoadRunner | 1-11 |
| Overview | 1-11 |
| Install New Network License Manager for RoadRunner Only | 1-11 |
| Install New Network License Manager for All Products | 1-13 |
| Update Network License Manager for RoadRunner | 1-16 |
| Overview | 1-16 |
| Update Existing Network License Manager for New RoadRunner Installation | 1-16 |
| Update Existing Network License Manager to Upgrade RoadRunner Software | 1-17 |
| Create Simple RoadRunner Scene | 1-19 |
| Prerequisites | 1-19 |
| Create New Scene and Project | 1-20 |
| Add Roads | 1-20 |
| Add Surface Terrain | 1-23 |
| Add Elevation and Bridges | 1-25 |
| Modify Junction | 1-28 |
| Add Crosswalk | 1-29 |
| Add Turning Lanes | 1-31 |
| Add Props | 1-37 |
| Other Things to Try | 1-42 |

| | |
|--|-------------|
| Camera Control in RoadRunner | 1-44 |
| Open Scene | 1-44 |
| Rotate Camera | 1-44 |
| Zoom Camera In and Out | 1-45 |
| Move Camera Horizontally | 1-46 |
| Move Camera Vertically | 1-47 |
| Frame Camera on Selected Object | 1-49 |
| Change View Projections | 1-51 |
| Set View Direction of Camera | 1-52 |
| | |
| Create Roads Around Imported GIS Assets | 1-55 |
| Download and Import GIS Assets into RoadRunner | 1-55 |
| Set World Origin | 1-56 |
| Add GIS Assets | 1-56 |
| Create Roads Around GIS Assets | 1-58 |
| Compare Roads Against Imported GIS Assets | 1-60 |

RoadRunner Fundamentals

2

| | |
|--|-------------|
| RoadRunner Project and Scene System | 2-2 |
| Projects | 2-2 |
| Scenes | 2-4 |
| Project and Scene Version Control | 2-4 |
| | |
| Window Layouts | 2-6 |
| Switch Between Tabbed Panes | 2-6 |
| Undock a Pane | 2-7 |
| Dock a Pane | 2-8 |
| Save the Current Window Layout | 2-8 |
| Restore a Saved Window Layout | 2-9 |
| Delete a Saved Window Layout | 2-9 |
| Reset the Window Layout to the Default Layout | 2-9 |
| | |
| Coordinate Space and Georeferencing | 2-10 |
| Local Coordinate System | 2-10 |
| Georeferencing (Geographic Coordinates and Projections) | 2-11 |
| | |
| Manipulate Scene Objects | 2-14 |
| Select Objects | 2-14 |
| Move Objects | 2-20 |
| Create Objects | 2-20 |
| Delete Objects | 2-21 |
| Modify Objects | 2-21 |
| | |
| Keyboard Shortcuts and Mouse Actions for RoadRunner | 2-23 |
| Scene Editing | 2-23 |
| Object Selection | 2-23 |
| Camera Control (Scene Editing Canvas) | 2-23 |
| Camera Control (2D Editor) | 2-24 |
| Scene Views | 2-25 |
| Utilities | 2-25 |

| | |
|---|-------------|
| File Operations | 2-25 |
| Update Linux Ubuntu Key Mapping | 2-25 |
| Choose a RoadRunner Tool | 2-27 |
| Road Tools | 2-27 |
| Junction Tools | 2-29 |
| Lane Tools | 2-29 |
| Marking Tools | 2-31 |
| Prop Tools | 2-32 |
| Terrain Tools | 2-33 |
| GIS Tools | 2-34 |
| Utility Tools | 2-35 |
| RoadRunner Asset Types | 2-37 |
| Texture and Material Assets | 2-37 |
| Prop Assets | 2-37 |
| Marking Assets | 2-39 |
| Road Assets | 2-40 |
| GIS Assets | 2-40 |
| Create, Import, and Modify Assets | 2-42 |
| Create and Import Assets | 2-42 |
| Modify Assets | 2-44 |
| Manage Assets | 2-44 |
| Visualize Assets | 2-45 |
| Add Asset to Scene | 2-47 |
| Resolve Geometry Issues | 2-48 |
| Angle Threshold | 2-48 |
| Show Edge Visualization | 2-49 |
| Detect Geometry Issues | 2-50 |
| Point Editing | 2-52 |
| Create a New Point | 2-52 |
| Move a Point | 2-52 |
| Curve Editing | 2-53 |
| Create a New Curve | 2-53 |
| Extend a Curve at Its Ends by Adding Control Points | 2-53 |
| Add Control Points to the Interior of a Curve | 2-54 |
| Move a Control Point | 2-54 |
| Change the Tangents of a Curve | 2-54 |
| Polygon Editing | 2-55 |
| Create a New Polygon | 2-55 |
| Add Control Points to a Polygon | 2-55 |
| Move a Control Point | 2-56 |
| Change the Tangents of a Polygon | 2-56 |
| Tangent Editing | 2-57 |
| Adjust a Tangent | 2-57 |
| Make Tangents Continuous | 2-58 |
| Make Tangents Discontinuous | 2-60 |

| | |
|--|-------------|
| Span Editing | 2-62 |
| Span Overview | 2-62 |
| Select a Span or Span Node | 2-63 |
| Create a New Span Node | 2-63 |
| Edit Attributes of a Span or Span Node | 2-64 |
| Move a Span Node | 2-64 |
| Delete a Span Node | 2-64 |
| Tips for Deleting Nodes | 2-64 |
| Region Graph Editing | 2-65 |
| Create a Graph Edge Curve | 2-65 |
| Split a Graph Edge Curve | 2-65 |
| Move a Graph Node | 2-66 |
| Change the Tangents of a Graph Edge Curve | 2-66 |
| Create a Region | 2-66 |
| Split a Region | 2-66 |
| Regions With Holes | 2-67 |
| Graphics and Startup Issues | 2-68 |
| System Requirements | 2-68 |
| Graphics Drivers | 2-68 |
| Laptops | 2-69 |
| Remote Desktops | 2-69 |
| Video Card Connection | 2-70 |
| Further Support | 2-71 |
| Obtain RoadRunner Log Files | 2-72 |
| Locate Log Folder | 2-72 |
| Provide Log File Contents to MathWorks Technical Support | 2-72 |

Import Data

3

| | |
|--|------------|
| Importing OpenDRIVE Files | 3-2 |
| OpenDRIVE Overview | 3-2 |
| Import OpenDRIVE into Scene | 3-2 |
| Import Objects, Signals, and Markings | 3-2 |
| Import Options | 3-2 |
| Limitations | 3-2 |
| Decompress LAZ Files | 3-4 |
| Decompression Process | 3-4 |
| GIS Data Resources for RoadRunner | 3-7 |
| Importing GIS Data in RoadRunner | 3-7 |
| USGS | 3-7 |

4

| | |
|--|------------|
| Resolve Triangulation Issues in Junctions | 4-2 |
| Adjust Road Elevations | 4-2 |
| Bank Roads | 4-2 |
| Use Slip Connections | 4-3 |
| How Surfaces Work in RoadRunner | 4-4 |
| Terrain Surface Model | 4-4 |
| Surfaces and Roads | 4-7 |
| Bridges | 4-9 |
| Extruded Surfaces | 4-10 |
| Surfaces and Elevation | 4-11 |

Export Scenes

5

| | |
|--|-------------|
| Export to AutoCAD | 5-2 |
| AutoCAD Export | 5-2 |
| Export to FBX | 5-3 |
| FBX Export | 5-3 |
| Advanced Details | 5-3 |
| Export to glTF | 5-5 |
| glTF Export | 5-5 |
| Limitations | 5-5 |
| Export to OpenFlight | 5-6 |
| OpenFlight Export | 5-6 |
| Limitations | 5-6 |
| Export to OpenSceneGraph | 5-7 |
| OpenSceneGraph Export | 5-7 |
| Limitations | 5-7 |
| Export to Wavefront OBJ | 5-8 |
| Wavefront Export | 5-8 |
| Advanced Details | 5-8 |
| Export to GeoJSON | 5-9 |
| GeoJSON Export | 5-9 |
| Export Options | 5-9 |
| Sample GeoJSON File | 5-9 |
| Traffic Signal Phases in GeoJSON | 5-11 |
| Export to USD | 5-14 |
| USD Export | 5-14 |
| Limitations | 5-14 |

| | |
|--|-------------|
| Export to OpenDRIVE | 5-15 |
| OpenDRIVE Overview | 5-15 |
| Export to OpenDRIVE | 5-15 |
| Export Options | 5-15 |
| OpenDRIVE Representations | 5-16 |
| OpenDRIVE User Asset Configuration | 5-22 |
| Limitations | 5-23 |
| Left-Hand Drive Export to OpenDRIVE 1.4 | 5-24 |
| Recommended Approach | 5-24 |
| OpenDRIVE Details | 5-24 |
| RoadRunner Export | 5-24 |
| Examples | 5-25 |
| Segmentation | 5-27 |
| Segmentation Overview | 5-27 |
| Toggle Segmentation Display | 5-27 |
| Categories | 5-27 |
| Downloading Plugins | 5-29 |
| Unity | 5-29 |
| Unreal and CARLA | 5-29 |
| RoadRunner Metadata Export | 5-30 |
| Metadata Overview | 5-30 |
| File Details | 5-30 |
| Export to Apollo | 5-32 |
| Apollo Overview | 5-32 |
| About the Different Apollo Maps | 5-32 |
| Generating Necessary Map Files | 5-32 |
| Visualizing Maps in Apollo Dreamview | 5-33 |
| Routing Simulations in Apollo Dreamview | 5-33 |
| Visualizing Maps in LGSVL | 5-33 |
| Apollo User Asset Configuration | 5-33 |
| Unsupported Features | 5-35 |
| Export to Metamoto | 5-36 |
| Export to Unity | 5-37 |
| Unity Overview | 5-37 |
| Installing the Import Tool | 5-37 |
| Exporting from RoadRunner to Unity | 5-39 |
| Importing into Unity | 5-41 |
| Setting Up the Sample Vehicle | 5-42 |
| Export to Unreal | 5-54 |
| Unreal Overview | 5-54 |
| Installing the Plugin | 5-54 |
| Exporting from RoadRunner to Unreal | 5-56 |
| Importing into Unreal | 5-57 |
| Importing Without the Plugin | 5-59 |
| Known Issues | 5-61 |

| | |
|--|-------------|
| Export to CARLA | 5-62 |
| CARLA Export Overview | 5-62 |
| Installing the Plugins | 5-62 |
| Exporting from RoadRunner to CARLA | 5-64 |
| Importing into CARLA/Unreal | 5-67 |
| Creating Map Packages for Distribution (CARLA 0.9.5) | 5-70 |
| Export to VTD | 5-72 |
| Exporting to VTD | 5-72 |
| Export Options (OpenDRIVE) | 5-72 |
| Export Options (OpenSceneGraph) | 5-73 |
| Import into VTD | 5-74 |
| Limitations | 5-76 |
| Customize Levels of Detail in Exported Scenes | 5-77 |
| Set Levels of Detail in Scene | 5-77 |
| Modify Triangulation Settings | 5-78 |
| Modify Scene Rendering | 5-82 |
| Pack Props | 5-84 |
| Visualize Performance Improvements | 5-86 |
| Export Scene | 5-87 |
| Export Custom Formats | 5-88 |
| Create Export Configuration XML File | 5-88 |
| Save Export Configuration File to Project | 5-92 |

RoadRunner Asset Library Product Overview

6

| | |
|---|------------|
| RoadRunner Asset Library Product Description | 6-2 |
|---|------------|

Get Started with RoadRunner

RoadRunner Product Description

Design 3D scenes for automated driving simulation

RoadRunner is an interactive editor that lets you design 3D scenes for simulating and testing automated driving systems. You can customize roadway scenes by creating region-specific road signs and markings. You can insert signs, signals, guardrails, and road damage, as well as foliage, buildings, and other 3D models. RoadRunner provides tools for setting and configuring traffic signal timing, phases, and vehicle paths at intersections.

RoadRunner supports the visualization of lidar point cloud, aerial imagery, and GIS data. You can import and export road networks using OpenDRIVE[®]. 3D scenes built with RoadRunner can be exported in FBX[®], glTF[™], OpenFlight, OpenSceneGraph, OBJ, and USD formats. The exported scenes can be used in automated driving simulators and game engines, including CARLA, VIRES VTD, NVIDIA DRIVE Sim[®], LGSVL, Baidu Apollo[®], Unity[®], and Unreal Engine[®].

RoadRunner Asset Library lets you quickly populate your 3D scenes with a large set of realistic and visually consistent 3D models. RoadRunner Scene Builder lets you automatically generate 3D road models from HD maps.

RoadRunner System Requirements

RoadRunner is an interactive editor that lets you design 3D scenes for simulating and testing automated driving systems. Before you install RoadRunner, check that your system meets these required specifications.

| Specification | Recommended | Minimum Requirement |
|------------------|--|--|
| Operating System | Windows®: Windows 10 x64 Linux®: Ubuntu® 16.04+ | Windows: Windows 10 x64 Linux: Ubuntu 16.04 |
| CPU | Intel® or AMD® x86-64 processor with four logical cores operating at 3.5 GHz or higher | Intel or AMD x86-64 processor operating at 2.5 GHz or higher |
| Memory | 16 GB | 8 GB |
| Video Card | NVIDIA® GTX 1060 3 GB | OpenGL® 3.2-compatible with 1 GB VRAM |
| Disk | SSD hard drive | 2 GB available disk space |

For more details on graphics card requirements and support with graphics issues, see “Graphics and Startup Issues” on page 2-68.

See Also

More About

- “Install and Activate RoadRunner” on page 1-4

Install and Activate RoadRunner

RoadRunner is an interactive editor that lets you design 3D scenes for simulating and testing automated driving systems.

The procedures in this topic are for a single RoadRunner computer installation or update/upgrade on Windows or Linux. These procedures can be performed by an individual license holder or by an end user or administrator with a network license.

Follow these procedures to install RoadRunner for the first time, update an installed release of RoadRunner, or upgrade an installed release of RoadRunner to a new release.

Network License Administrators Before you or your end users install RoadRunner, perform the following tasks:

- Install the network license manager. See “Install Network License Manager for RoadRunner” on page 1-11.
 - Download the network license file and the platform-specific product installer and save them to removable media or a network location. You will need to give these items to your end users for them to install on their own computers.
-

Verify System Requirements

Before you install RoadRunner, check that your system meets these required specifications.

| Specification | Recommended | Minimum Requirement |
|------------------|--|--|
| Operating System | Windows: Windows 10 x64 Linux: Ubuntu 16.04+ | Windows: Windows 10 x64 Linux: Ubuntu 16.04 |
| CPU | Intel or AMD x86-64 processor with four logical cores operating at 3.5 GHz or higher | Intel or AMD x86-64 processor operating at 2.5 GHz or higher |
| Memory | 16 GB | 8 GB |
| Video Card | NVIDIA GTX 1060 3 GB | OpenGL 3.2-compatible with 1 GB VRAM |
| Disk | SSD hard drive | 2 GB available disk space |

Get License and Product Installer

Use an Individual License

Get your RoadRunner license by following these steps.

- 1 Go to the License Center on the MathWorks® website.

If prompted, sign in to your MathWorks Account.

- 2 Click the RoadRunner license in your account.

- 3 On the Install and Activate tab, click **Activate a Computer**.
- 4 In the form displayed, enter all requested information. When finished, click **Submit**.
- 5 Download or email the license file and save it to a folder on the computer where you will install RoadRunner.

Next, download the product installer by following these steps.

- 1 Sign in to your MathWorks Account on the MathWorks website.
- 2 Under **My Software**, click the down arrow next to your RoadRunner license.

This action takes you to the Downloads page where you can find the download RoadRunner link. If you see **Get Latest Release**, click **Download R2021a** to get to the right page..

- 3 Under **Additional Product Downloads**, click **Get R2021a RoadRunner**.
- 4 Download the platform-specific installer to the computer on which you want to install RoadRunner.

Use a Network License

- **End Users** — Get the license file and platform-specific installer from your license administrator and copy them to the computer where you are installing RoadRunner.
- **Network/License Administrators** — Give end user the platform-specific installer (or provide access on a network share) and the modified network license. See “Step 3. Configure Network License” on page 1-12 for instructions on configuring the network license for end user access to RoadRunner software.

Install RoadRunner

For RoadRunner installation, GUI installation is available for Windows and GUI and command line methods are available on Linux platforms. Installation instructions vary based on the platform and method you choose.

| Platform and Installation Method | Instructions |
|----------------------------------|---|
| Windows GUI | <ol style="list-style-type: none"> 1 Double-click the downloaded product installer. 2 Follow all prompts to complete installation. |
| Linux GUI | <ol style="list-style-type: none"> 1 Double-click the <code>.deb</code> file that you just downloaded. 2 Click Install and follow all prompts. |
| Linux command line | <ol style="list-style-type: none"> 1 Open a command prompt. 2 Enter the following command, replacing <code>release_number</code> with the current release number: <pre>sudo dpkg -i RoadRunner/ release_number.deb</pre> <p>For example, for the R2021a release, use this command:</p> <pre>sudo dpkg -i RoadRunner_R2021a_glnxa64.deb</pre> 3 Follow all prompts to complete installation. |

Activate License

Before you can use RoadRunner, you must activate the license. Follow these steps to install an individual or network RoadRunner license on your computer.

Note If you do not have a license, see “Get License and Product Installer” on page 1-4.

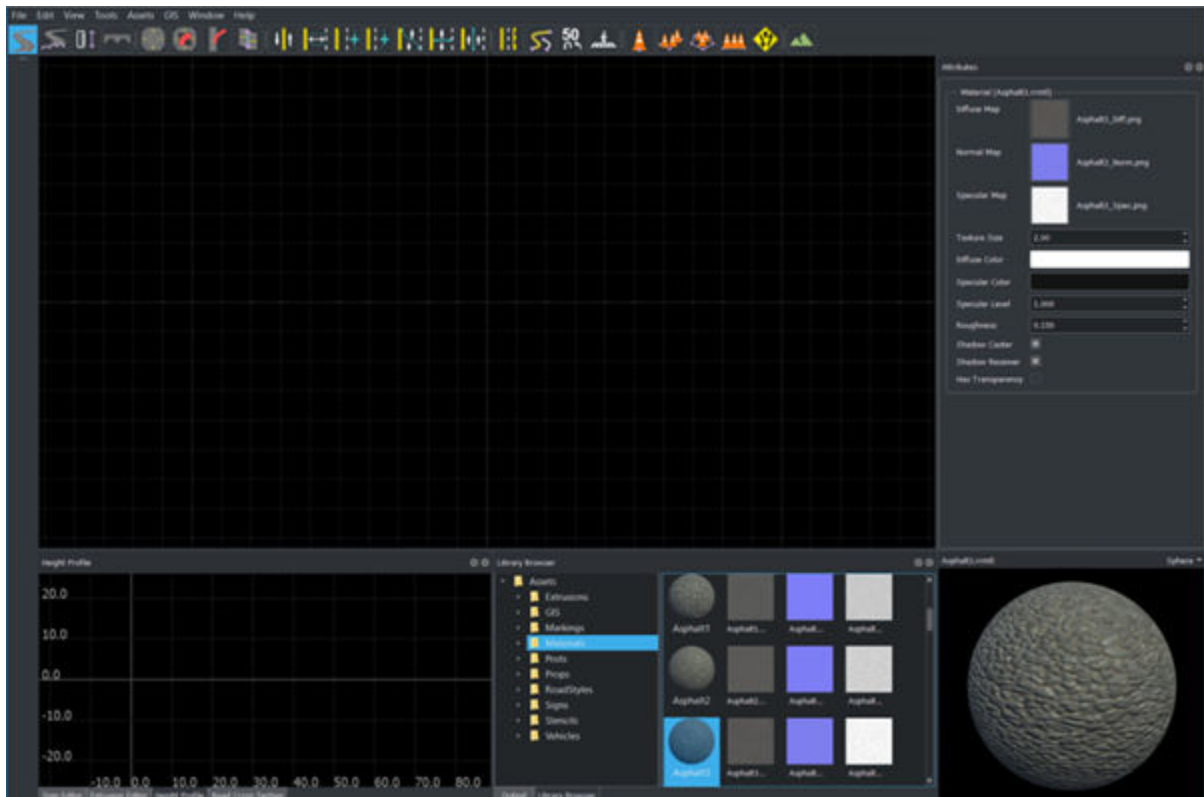
- 1** On your desktop, launch RoadRunner and follow all prompts.
 - **Windows** — Launch RoadRunner from the shortcut on the desktop or Start Menu.
 - **Linux** — Launch RoadRunner from the installed shortcut. To find this shortcut, click Home and type "roadrunner" in the text field or go to "/usr/share/applications".
- 2** When prompted for a license file, enter the path to the license file from “Get License and Product Installer” on page 1-4.
- 3** Follow any additional prompts to complete activation.

Create a New Project and Scene

After you install and activate RoadRunner, you can now create a project to get started creating scenes.

RoadRunner uses a project folder to store the assets (materials, models, and so on) that you can use within the application. If you do not already have an existing project, you need to create one. See “RoadRunner Project and Scene System” on page 2-2.

After following the previous instruction, RoadRunner opens to a new blank scene.



To get started, you can use the **Road Plan Tool** to create your first road. Alternatively, use one of the other tools to start creating your scene. For more on getting started, see the Getting Started with RoadRunner video series.

If you experience rendering issues or crashes on startup, see “Graphics and Startup Issues” on page 2-68 for troubleshooting help.

See Also

Related Examples

- “Get RoadRunner Updates and Upgrades” on page 1-8
- “Graphics and Startup Issues” on page 2-68
- “Import Data”
- “Design Scenes”
- “Choose a RoadRunner Tool” on page 2-27
- “Install Network License Manager for RoadRunner” on page 1-11
- “Update Network License Manager for RoadRunner” on page 1-16

External Websites

- License Center
- Getting Started with RoadRunner

Get RoadRunner Updates and Upgrades

Update Installed Release

You can download and install a more recent version of the release of MATLAB® that you currently have installed. For example, if you have R2021a installed, you might be able to download a more recent version of R2021a. This version is called a MATLAB update. When you install a MATLAB update, you get not only the most recent update for your version but also all previous updates for your version as well.

When an updated version of RoadRunner for your installed release is available, download the installer again, and then rerun the installation, following the instructions in “Install and Activate RoadRunner” on page 1-4.

Upgrade RoadRunner Release

Upgrading involves installing a newer general release than the current installation on your computer, for example, from R2020b to R2021a. The Software Maintenance Service on your license must be current and it determines which releases you can upgrade to. For more information, contact Sales.

To upgrade to a new release of RoadRunner, rerun the steps in “Install and Activate RoadRunner” on page 1-4.

Install RoadRunner Add-On Products

To install RoadRunner add-on products, for example, the RoadRunner Asset Library and RoadRunner Scene Builder, complete the following steps.

- 1 Contact Sales to put the add-on product on your RoadRunner license.
- 2 Follow all instructions in “Install and Activate RoadRunner” on page 1-4. You must get an updated license, download and rerun the installer to get the added products, and activate the updated license.

See Also

Related Examples

- “Install and Activate RoadRunner” on page 1-4
- “Update RoadRunner Licenses” on page 1-9

Update RoadRunner Licenses

If you are an individual RoadRunner user, you can download and install an updated RoadRunner license on your computer. If you are a network license administrator, you can update the network license for add-on products or to add users or seats to the license.

Update RoadRunner Individual License

When you get an updated license for RoadRunner, you must update the license on your computer.

To update your license, use the following procedure.

- 1 Go to the License Center on the MathWorks website. Sign in to your MathWorks account if prompted.
- 2 Click the RoadRunner license in your account.
- 3 On the Install and Activate tab, under **Get License File**, click the down arrow next to your RoadRunner license.
- 4 Enter the release for the license, and then click **Continue**.
- 5 Download or email the license file and save it to a folder on the computer where the previous RoadRunner license was located. When you have completed this step, click **Done**.
- 6 Launch RoadRunner, and then click **License** to update the license on your computer.
- 7 When prompted for the license file, enter the path to the license file you just downloaded.
- 8 Follow any additional prompts to complete the updated license activation.
- 9 Restart RoadRunner.

Update RoadRunner Network Licenses

When you receive a new license from MathWorks because the products or seat counts have changed, plan to update the license file on the network server at a time when users are least likely to be accessing a RoadRunner license.

To update the license:

- 1 Make a copy of the existing RoadRunner network license on the license server.
- 2 Go to License Center on the MathWorks website. Sign in to your MathWorks Account, if prompted.
- 3 On the Install and Activate tab, under **Get License File**, click the down arrow next to RoadRunner Server.
- 4 Download or email the license file and save it to a folder on the computer where the previous RoadRunner network license is located. When you have completed this step, click **Done**.
- 5 Stop the network license manager.
- 6 Open the existing license on the server and the new license in an editor. You are going to copy most of the new license into the old license with these instructions:
 - a In the existing license, delete all content except the **SERVER** and **DAEMON** lines at the top of the file.
 - b In the new license, copy all content starting from below the **SERVER** and **DAEMON** lines to the end.

- c** Paste the copied content into the existing license below the SERVER and DAEMON lines.
 - d** Save the existing license. You can store the new license as a backup, remembering that the SERVER and DAEMON lines must be replaced with those specific to your organization.
- 7** If you have an options file, depending on how it is configured, you might have to update it. If you do not have an options file, skip this step.
- 8** Start the network license manager.
- 9** Have end users restart RoadRunner.

See Also

Related Examples

- “Update Network License Manager for RoadRunner” on page 1-16

Install Network License Manager for RoadRunner

Overview

Note You must be a **network license administrator** to perform these procedures. If you are an end user on a network license and have been asked to install and/or activate RoadRunner yourself, follow the instructions in “Install and Activate RoadRunner” on page 1-4.

Before you or your end users can install and run RoadRunner software, you must first install a network license manager from MathWorks to manage the RoadRunner network license. The instructions in this topic apply to both Network Named User and Concurrent licenses.

Select a workflow from the following table.

| Goal | Procedure |
|--|---|
| You want to install a network license manager for RoadRunner only. | “Install New Network License Manager for RoadRunner Only” on page 1-11. |
| You want to install a network license manager for RoadRunner and other MathWorks products. | “Install New Network License Manager for All Products” on page 1-12. |

If you already have the MathWorks network license manager and you want to add RoadRunner, see “Update Existing Network License Manager for New RoadRunner Installation” on page 1-16.

Install New Network License Manager for RoadRunner Only

Follow this workflow if you are installing the network license manager from MathWorks for the first time and you are planning to install only RoadRunner at this time.

Step 1. Download RoadRunner Network License

- 1 Go to License Center on the MathWorks website and select the RoadRunner network license.
- 2 On the Install and Activate tab, click **Activate to Retrieve License File** and follow all prompts. For application label, enter "RoadRunner Server."
- 3 Download or email the license file and save it on the server that will be hosting the network license manager.

Step 2. Install Network License Manager

- 1 Download the MathWorks-specific license manager daemons from License Manager Files on the MathWorks website.

The license manager consists of four binaries:

- Imgrd, the core license manager binary
- MLM, the MATLAB vendor daemon
- lmutil, a suite of tools for administering the license manager
- lmttools.exe, a graphical front end for the license manager (Windows only).

- 2 Manually install the network license manager:
 - a Extract the downloaded folder and place the folder in the desired destination
 - b Place the file `license.lic` (downloaded in the previous step) somewhere you can remember, optimally with `lmgrd` and `MLM`.

When you have completed installing the network license manager, go to the next step, Configure Network License. Do not start the network license manager at this time.

Step 3. Configure Network License

For all network license types, configure network and end user license. If using a Network Named User License, set up named users.

- 1 Open the network license file for editing.
- 2 The `SERVER` line identifies the server (host and port number). Add a `SERVER` line to the top of the license file, as follows:

```
SERVER host hostid port
```

An example of this syntax is:

```
SERVER Server1 0123abcd0123 12345
```

- 3 The `DAEMON` line identifies the name of the network license manager daemon. Add the `DAEMON` line with the name of the network license manager daemon to the next line, as follows:

```
DAEMON MLM <path to MLM.exe>
```

An example of this syntax is:

```
DAEMON MLM $lmroot/etc/glnxa64/MLM
```

`$lmroot` is where you installed the network license manager.

- 4 Create end user license.
 - a Create a license file that points to the license server for each end user installation of RoadRunner.
 - i Create a text file, and name it `network.lic`.
 - ii From the network license file, copy the `SERVER` line and paste it into the new license file as the first line. This server line specifies the host, hostID, and port of the license server and has the following format:

```
SERVER host hostid port
```

An example of this syntax is:

```
SERVER Server1 0123abcd0123 1711
```

By copying the line rather than recreating it, you are less likely to mistype the information.

- iii For the second line, add:

```
USE_SERVER
```

- iv Save the license file.
- b Put the `network.lic` file on each end user computer, using one of two options.
 - Option 1: If the end user will be performing the activation, give the license file to them.
 - Option 2: On the end user's computer, put the license file in an accessible folder and set the "MLM_LICENSE_FILE" environment variable to specify the path to the file (or the folder containing it).
- 5 If you are the administrator on a Network Named User License, complete the following steps:
 - a Follow the procedure in "Set Up Named User Licensing", in which you will create an options file. This options file is used by the network license manager to identify the specific named users to whom you have assigned right-to-use privileges.
 - b Make sure the DAEMON line in the license file you downloaded during network license installation contains the correct path to the options file. See "Check the Options File".

Step 4. Start Network License Manager

Start the license manager.

Step 5. Ready to Install RoadRunner

When you have completed updating the network license manager, you can then install RoadRunner software on individual computers.

You have the following options for installing RoadRunner:

- You can install and activate each installation yourself. See "Install and Activate RoadRunner" on page 1-4 and follow all steps in the procedure.
- You can have each end user perform their own installation and activation. Give each user a copy of the network license file (`network.lic`) and the platform-specific product installer and have them follow all steps in "Install and Activate RoadRunner" on page 1-4. You must set up the network license first; see Set up Network License
- You can install RoadRunner on each computer but have the end user activate it.
 - 1 "Install and Activate RoadRunner" on page 1-4 (but don't activate the software)
 - 2 Give the end user a copy of the network license (`network.lic`) and have them put it on the end user computer where RoadRunner was installed.
 - 3 Instruct the end user to follow the procedures in "Activate License" on page 1-6.

Note If you are performing a RoadRunner installation for the first time, consider first installing and activating RoadRunner on a test computer. If your test is successful, you can start the individual installations with confidence.

Install New Network License Manager for All Products

Follow this procedure if you are planning to install RoadRunner and other MathWorks products for the first time, but you have not yet installed the network license manager.

Step 1. Install Network License Manager and Other MathWorks Products

For this procedure, before you add the RoadRunner license, you will use the MathWorks product installer to install the network license manager and other MathWorks products.

- 1 Select one of the following options to install the network license manager:
 - “Install Network License Manager with Internet Connection”
 - “Install Network License Manager Offline”
- 2 Go to “Install Products” and complete your MATLAB installation. When you have finished, return here.

Step 2. Download and Configure Network Licenses

- 1 If the network license manager is running, Stop the license manager before continuing.
- 2 Go to License Center on the MathWorks website and select the RoadRunner network license.
- 3 On the Install and Activate tab, click **Activate to Retrieve License File** and follow all prompts. For application label, enter "RoadRunner Server."
- 4 Download or email the license file and save it on the server that is hosting the network license manager.
- 5 Combine your RoadRunner network license with the MATLAB network license. See How do I serve multiple MATLAB licenses from the same network license manager? in MATLAB Answers.

Step 3. Start Network License Manager

Start the license manager.

Step 4. Ready to Install RoadRunner

When you have completed updating the network license manager, you can then install RoadRunner software on individual computers.

You have the following options for installing RoadRunner:

- You can install and activate each installation yourself. See “Install and Activate RoadRunner” on page 1-4 and follow all steps in the procedure.
- You can have each end user perform their own installation and activation. Give each user a copy of the network license file (`network.lic`) and the platform-specific product installer and have them follow all steps in “Install and Activate RoadRunner” on page 1-4. You must set up the network license first; see Set up Network License
- You can install RoadRunner on each computer but have the end user activate it.
 - 1 “Install and Activate RoadRunner” on page 1-4 (but don't activate the software)
 - 2 Give the end user a copy of the network license (`network.lic`) and have them put it on the end user computer where RoadRunner was installed.
 - 3 Instruct the end user to follow the procedures in “Activate License” on page 1-6.

Note If you are performing a RoadRunner installation for the first time, consider first installing and activating RoadRunner on a test computer. If your test is successful, you can start the individual installations with confidence.

See Also

Related Examples

- “Install and Activate RoadRunner” on page 1-4
- “Update RoadRunner Network Licenses” on page 1-9
- “Get RoadRunner Updates and Upgrades” on page 1-8
- “Update Network License Manager for RoadRunner” on page 1-16

More About

- “Network Named User License Administration”
- “Concurrent License Administration”

External Websites

- [License Center](#)

Update Network License Manager for RoadRunner

Overview

Follow the procedures in this topic if you are adding or upgrading RoadRunner software and require an updated network license manager from the same release.

- If you are adding RoadRunner, update the network license manager first.
- If you are upgrading RoadRunner to a newer release, you can perform the network license manager update and the RoadRunner upgrade in either order.

Either way, you must have the new or updated license in place before you restart the network license manager. Instructions are in the procedures described in this topic.

Note You must be a network license administrator to perform these procedures. Updating the network license manager software requires you to stop and restart the license manager.

Select a workflow from the following table.

| Goal | Procedure |
|--|---|
| You want to install RoadRunner with an existing network license manager from MathWorks. | "Update Existing Network License Manager Installation" on page 1-16 |
| You want to upgrade RoadRunner to a newer release, and need to update the network license manager to the same release. | "Update Existing Network License Manager Software" on page 1-17 |

Update Existing Network License Manager for New RoadRunner Installation

To add RoadRunner to an existing MATLAB installation, the network license manager version must be the same or higher as the version of RoadRunner you plan to install. For example, if your MathWorks products and network license manager are from R2020a and you want to download and install RoadRunner R2020b, you must update your existing network license manager to the R2020b release.

Step 1. Download RoadRunner Network License

- 1 Go to License Center on the MathWorks website and select the RoadRunner network license.
- 2 On the Install and Activate tab, click **Activate to Retrieve License File** and follow all prompts. For application label, enter "RoadRunner Server."
- 3 Download or email the license file and save it on the server that is hosting the network license manager.

Step 2. Configure Network License

- 1 Combine your new RoadRunner network license with the existing MATLAB network license. See How do I serve multiple MATLAB licenses from the same network license manager? in MATLAB Answers.
- 2 If you have a Network Named User license and are adding users for RoadRunner, modify the options file to include the additional users.

- 3 If you have a Concurrent license and have added new seats for RoadRunner, create a license file for each new machine that will be running RoadRunner by copying the license file you already created for your other RoadRunner installations..

Step 3. Update Network License Manager

Follow the instructions in “Update Network License Manager Software”. This procedure requires you to stop and restart the network license manager.

Step 4. Ready to Install RoadRunner

When you have completed updating the network license manager, you can then install RoadRunner software on individual computers.

You have the following options for installing RoadRunner:

- You can install and activate each installation yourself. See “Install and Activate RoadRunner” on page 1-4 and follow all steps in the procedure.
- You can have each end user perform their own installation and activation. Give each user a copy of the network license file (`network.lic`) and the platform-specific product installer and have them follow all steps in “Install and Activate RoadRunner” on page 1-4. You must set up the network license first; see Set up Network License
- You can install RoadRunner on each computer but have the end user activate it.
 - 1 “Install and Activate RoadRunner” on page 1-4 (but don't activate the software)
 - 2 Give the end user a copy of the network license (`network.lic`) and have them put it on the end user computer where RoadRunner was installed.
 - 3 Instruct the end user to follow the procedures in “Activate License” on page 1-6.

Note If you are performing a RoadRunner installation for the first time, consider first installing and activating RoadRunner on a test computer. If your test is successful, you can start the individual installations with confidence.

Update Existing Network License Manager to Upgrade RoadRunner Software

Follow these procedures if you are updating the network license manager so that you can upgrade your RoadRunner release.

- 1 If you have received an updated RoadRunner network license, complete the steps in “Update RoadRunner Licenses” on page 1-9 first.
- 2 Stop the license manager.
- 3 Download the MathWorks-specific license manager daemons from License Manager Files on the MathWorks website.
- 4 Manually install the network license manager by extracting the downloaded folder and placing the folder where you had the previous network manager files.
- 5 Start the license manager.

You can now upgrade the software. See “Upgrade RoadRunner Release” on page 1-8.

See Also

Related Examples

- “Install and Activate RoadRunner” on page 1-4
- “Update RoadRunner Network Licenses” on page 1-9
- “Get RoadRunner Updates and Upgrades” on page 1-8

More About

- “Network Named User License Administration”
- “Concurrent License Administration”

External Websites

- [License Center](#)

Create Simple RoadRunner Scene

In this section...

“Prerequisites” on page 1-19
“Create New Scene and Project” on page 1-20
“Add Roads” on page 1-20
“Add Surface Terrain” on page 1-23
“Add Elevation and Bridges” on page 1-25
“Modify Junction” on page 1-28
“Add Crosswalk” on page 1-29
“Add Turning Lanes” on page 1-31
“Add Props” on page 1-37
“Other Things to Try” on page 1-42

RoadRunner is an interactive editor that lets you design 3D scenes for simulating and testing automated driving systems. This example shows how to create a simple scene containing an intersection, bridges, and trees in the surrounding terrain, similar to the scene shown here:



Prerequisites

Before beginning this example, make sure that your system meets these prerequisites:

- You have downloaded, installed, and activated RoadRunner by following the instructions described in “Install and Activate RoadRunner” on page 1-4.
- You have a license for the “RoadRunner Asset Library Add-On”. This example uses assets that are available only in this library.

Although this example covers some basic camera operations, for a more complete understanding of how the RoadRunner camera works, consider reviewing the “Camera Control in RoadRunner” on page 1-44 example first.

Create New Scene and Project

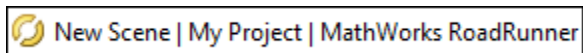
In RoadRunner, each scene you create is part of a project, which is a folder of assets (scene components) that can be shared across all scenes in that project. Create a new scene and a new project in which to put that scene.

- 1 Open RoadRunner, and from the start page, click **New Scene**.
- 2 On the Select a Project window, click **New Project**.
- 3 In your file system, browse for an empty folder in which to create the project. If an empty folder does not exist, create one and name it **My Project**. The folder name becomes the name of the project.
- 4 When prompted, click **Yes** to install the RoadRunner Asset Library in your project.

RoadRunner opens to a new scene with an empty scene editing canvas.




The name of the project that you specified appears in the title bar. The name of the scene also appears in the title bar, but it is displayed as **New Scene** until you save the scene and give it a name.

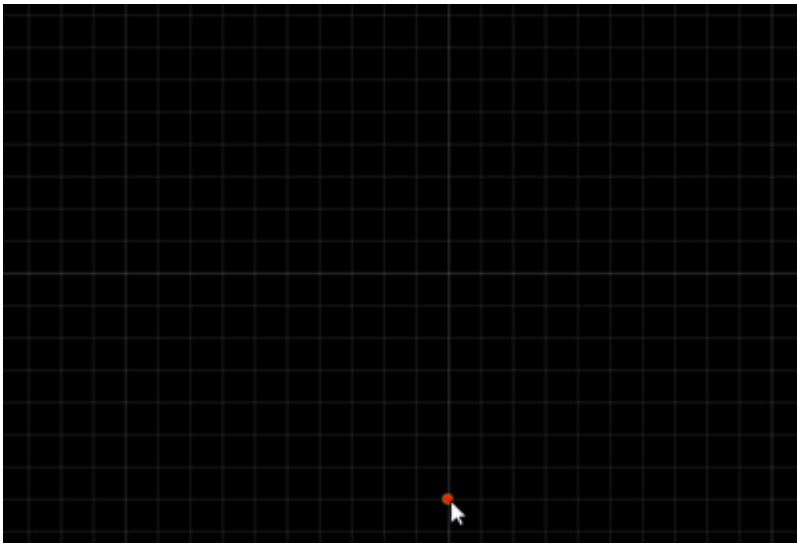


You can create a new scene, change scenes, or change projects at any time from the **File** menu. When you reopen RoadRunner, you can select recent scenes that you worked on from the start page, in the **Recent Scenes** list.

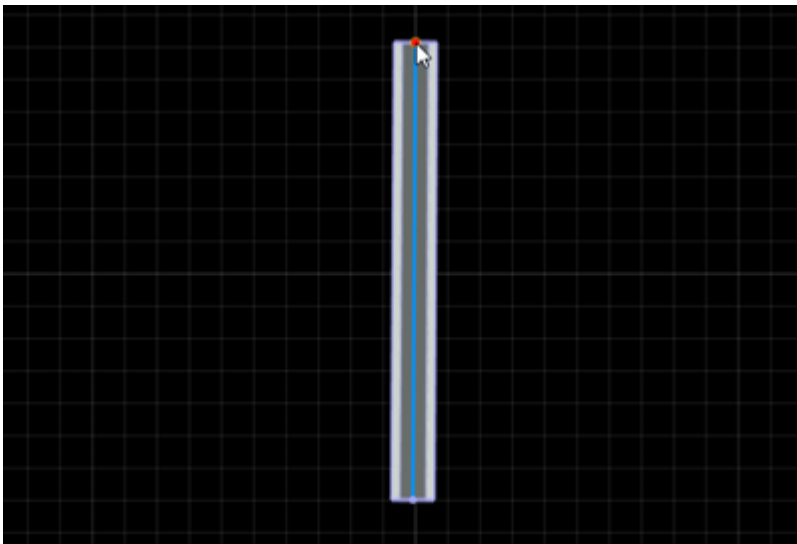
Add Roads

When you open a new scene, RoadRunner opens with the **Road Plan Tool**  selected. Instructions on using this tool appear in the bottom status bar. By right-clicking in the scene editing canvas with this tool selected, you can add control points that shape the geometry of a road.

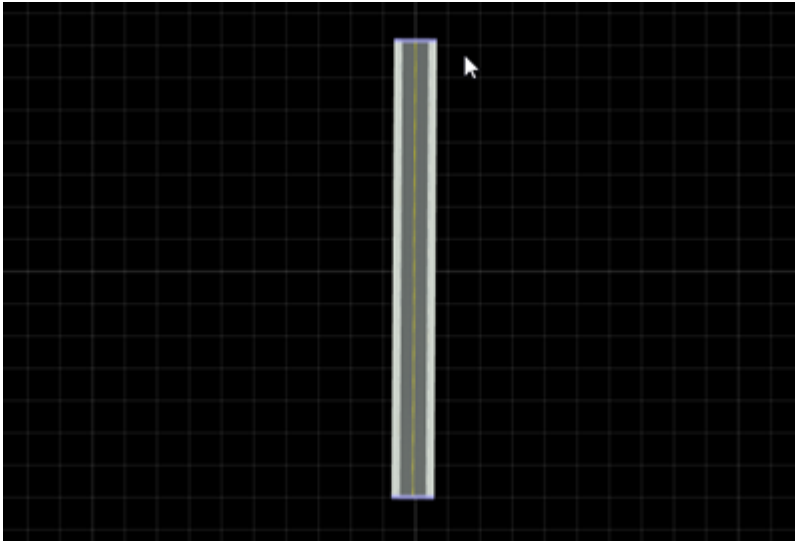
- 1 At the bottom center of the scene editing canvas, right-click to add the first control point of a new road.



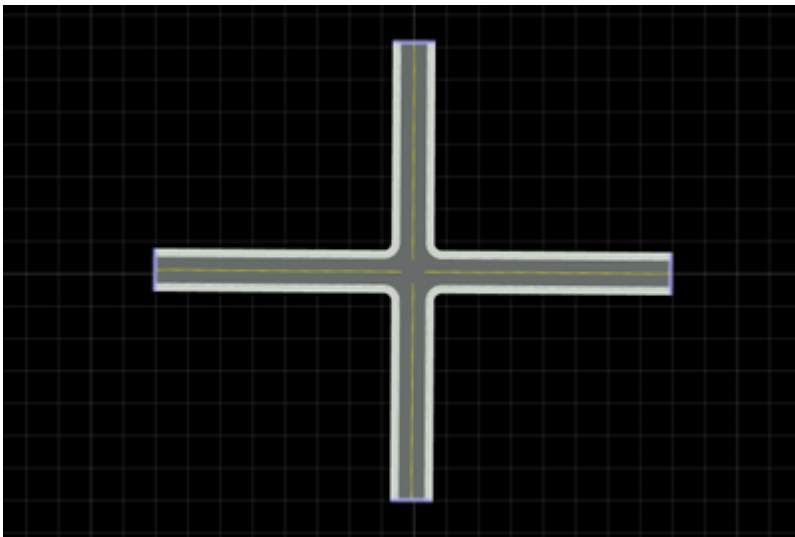
- 2 At the top center of the canvas, right-click to add a second control point and form your first road segment.



- 3 Click away from the road to deselect the road and finish creating it.

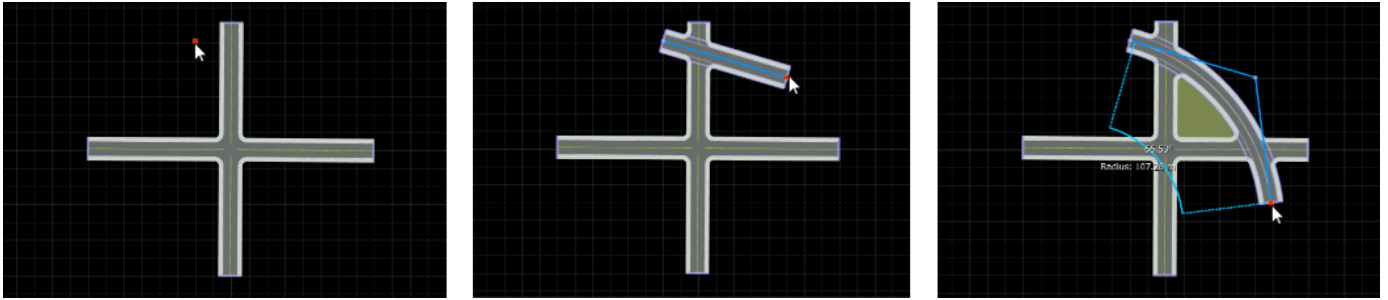


- 4** Create a new straight road that intersects the first road by right-clicking to its left, right-clicking to its right, and then clicking away from the road. The two roads form a junction.



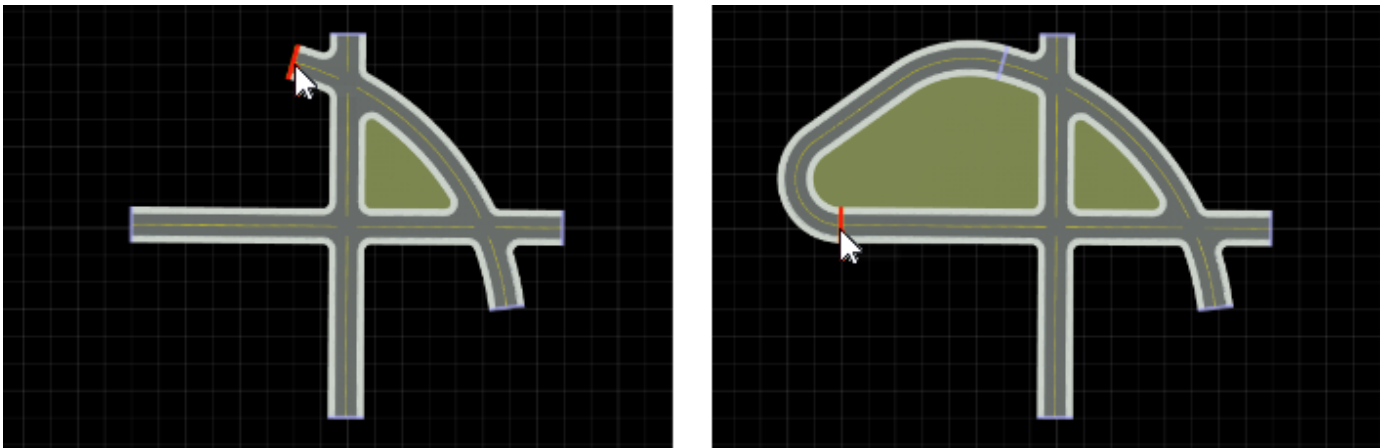
So far you have created straight roads. To form curved roads, right-click multiple times to add additional control points to a road. Create a curved road that overlaps the intersection.

- 1** Right-click within the top-left quadrant of the intersection.
- 2** Right-click within the top-right quadrant of the intersection. The first created road segment is straight.
- 3** Right-click in the bottom-right quadrant of the intersection. The area enclosed within the intersection and the curved road forms a ground surface.



You can extend existing roads by selecting the endpoint of a road and right-clicking to add more control points.


- 1 In the curved road you created, click to select the end near the top of the canvas.
- 2 Right-click the left end of the intersection. RoadRunner creates a road that meets the necessary geometric constraints. The enclosed area again forms a ground surface.



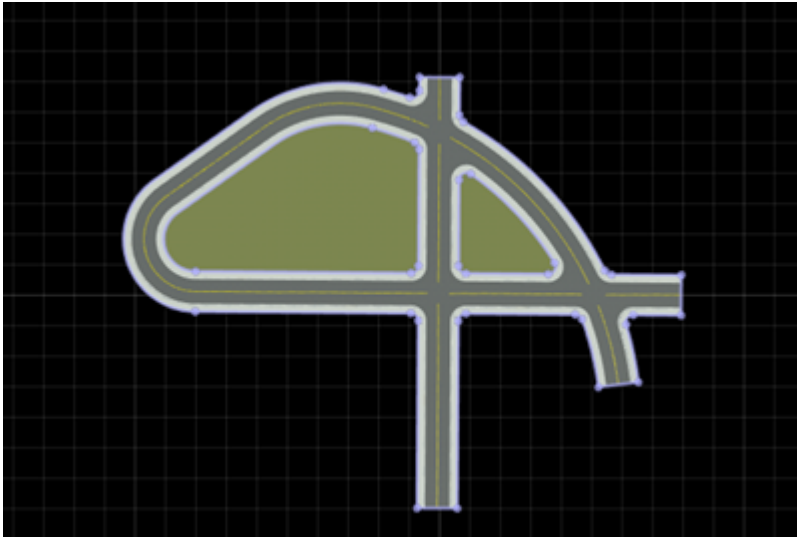
To modify any road, click to select it and try dragging its control points or moving the entire road. You can also right-click a road to add additional control points. For example, in this road network, you can add control points to smooth out the curve on the left side of the intersection.

Add Surface Terrain

So far, only the areas enclosed by roads contain surface terrain. To add surface terrain around the

entire road network, you can use the **Surface Tool** .

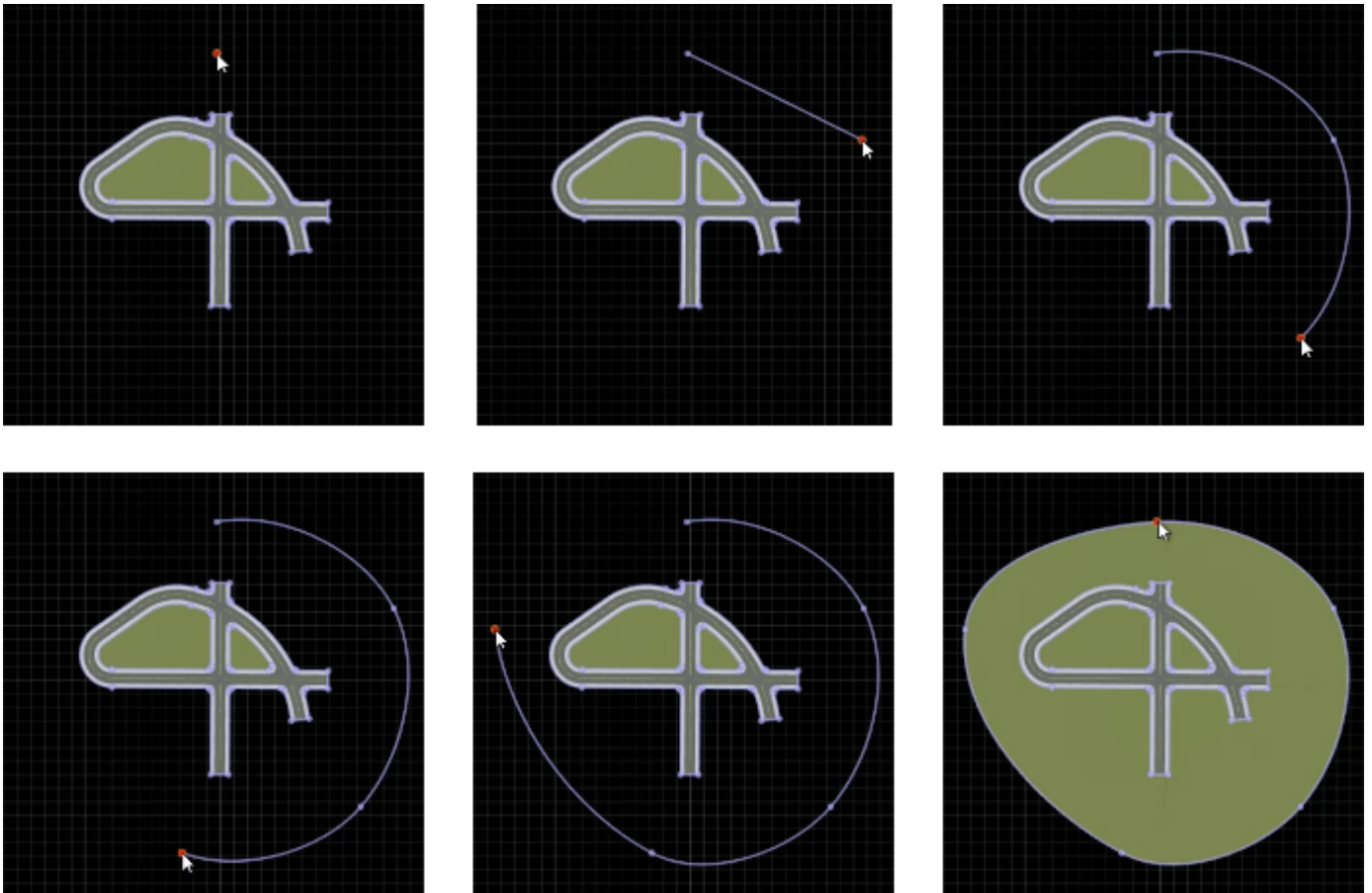
- 1 In the toolbar, click the **Surface Tool** button. Selecting a new tool puts RoadRunner in a different mode that enables new interactions and makes different scene objects selectable. With the **Surface Tool** selected, the roads are no longer selectable but the road surface nodes become selectable.



- 2 Zoom out of the scene, either by using the scroll wheel or by holding **Alt** and right-click and then dragging down or left.



- 3 Right-click above the road network to add a new surface node. Then, keep right-clicking at points around the road to form a circle. When you reach the top node again, right-click it to connect the surface graph and commit the surface to the canvas.

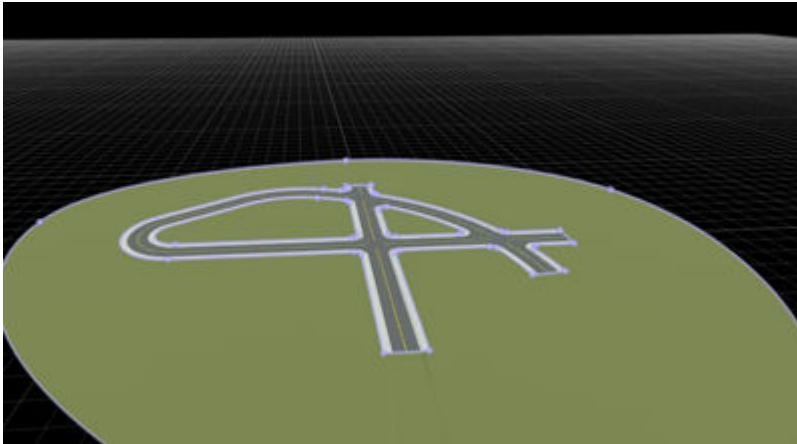


To modify the surface size, click and drag the surface nodes. To modify the curve of the surface, click the segments between nodes, and then click and drag the tangent lines.

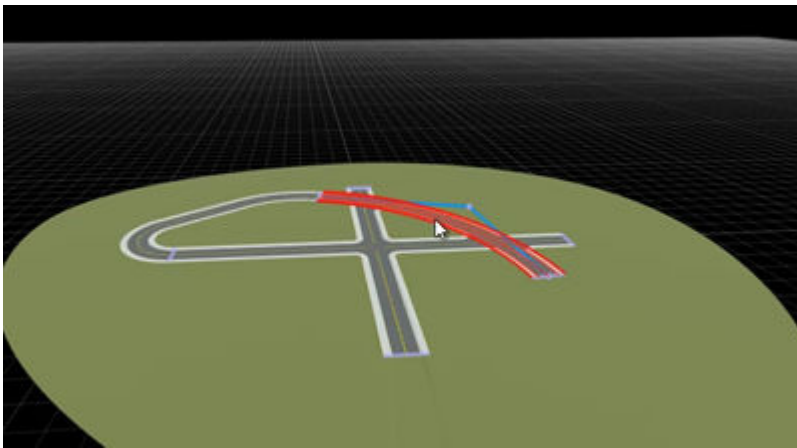
Add Elevation and Bridges

Up to this point, the scene has been flat. Modify elevation in the scene by changing the height of one of the roads.

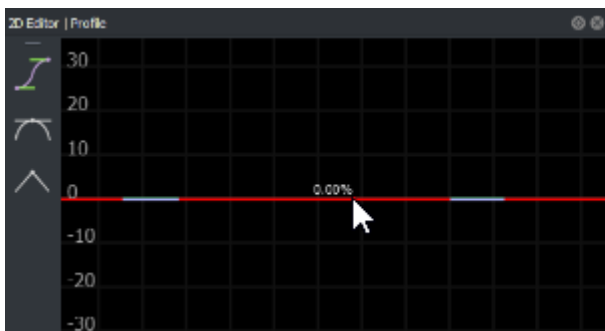
- 1 Hold **Alt** and then click and drag the camera to view the scene at an angle.



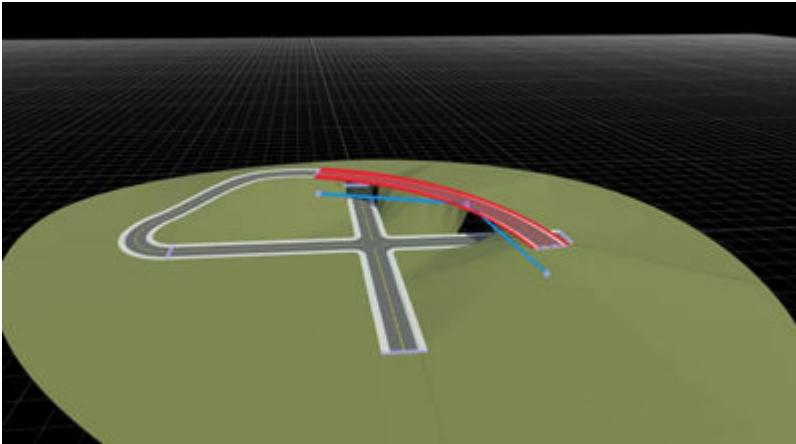
- 2 Click the **Road Plan Tool** button to make roads selectable again. Then, click to select the first curved road that you created.



- 3 To elevate the road, use the **2D Editor**, which enables you to view scene aspects such as road profiles and road cross-sections. In the **2D Editor**, select the road profile and raise it approximately 10 meters.



The road is now elevated in the scene canvas above the intersection. Instead of forming junctions, the elevated road forms overpasses.

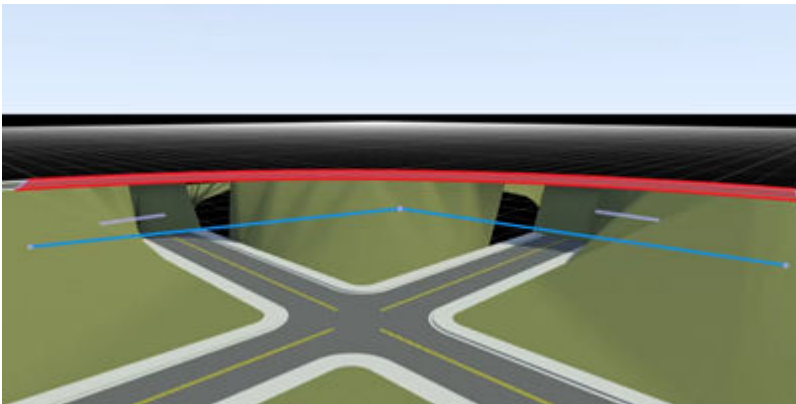


Roads attach to the surface terrain. When you elevate a road, the terrain elevates with it. Increasing elevation can lead to visual artifacts below the overpasses. To resolve this issue, you can create

bridge spans by using the **Road Construction Tool**




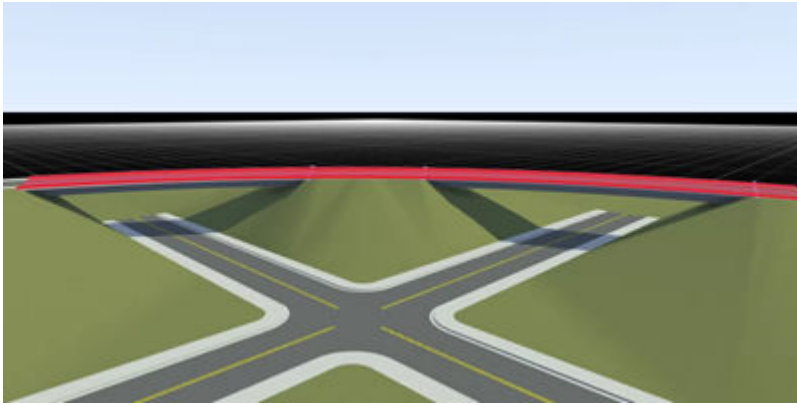
- 1 Rotate the camera and zoom in to see the visual artifacts at the overpasses.



- 2 Click the **Road Construction Tool** button.

3

On the left toolbar, click the **Auto Assign Bridges** button . This operation, which is available only when you are using the **Road Construction Tool**, converts only those road sections that are directly above a region to bridge spans. Use the default bridge span inflation and click **OK**. The roads spans are converted to bridges and the visual artifacts are removed.



If the bridges do not form correctly, try adjusting the road elevation or the bridge span inflation and rerun the **Auto Assign Bridges** operation.

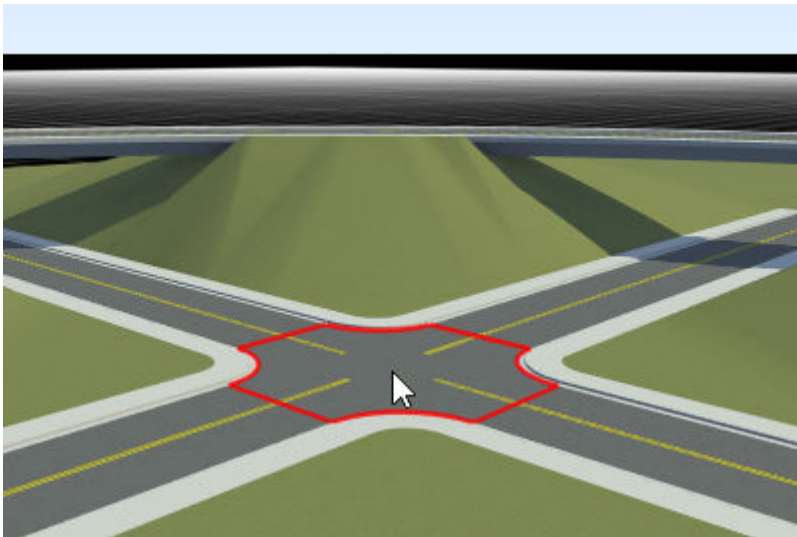
Modify Junction

Some tools enable you to select and modify properties at junctions. Modify the corner radius of the four-way intersection.

1

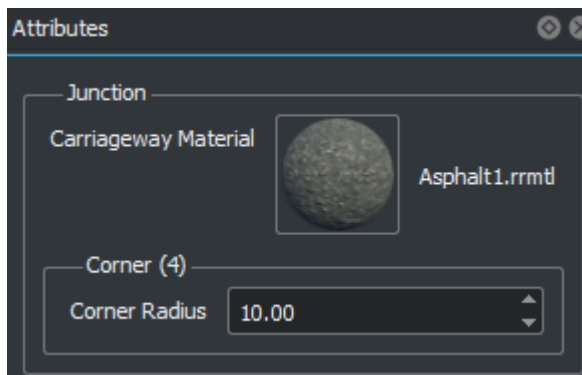


Click the **Corner Tool** button , and then click to select the four-way intersection.

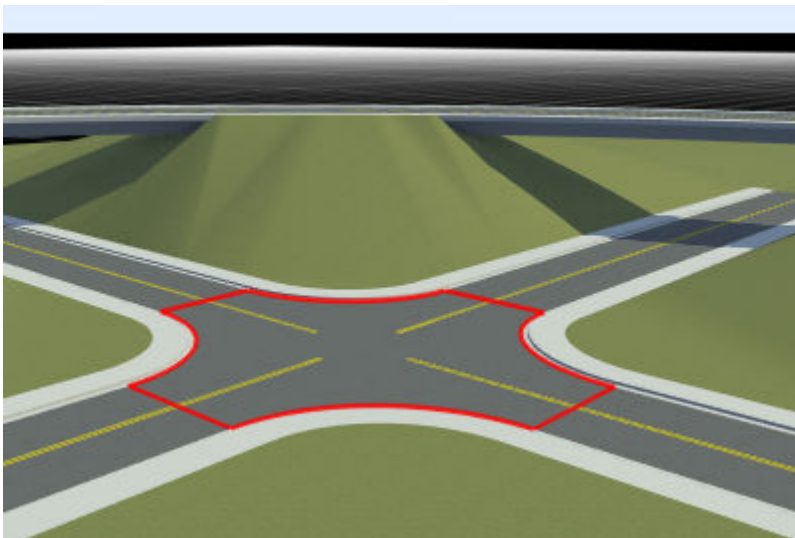


2 By default the junction has a corner radius of 5 meters. Increase this value by using the **Attributes** pane. This pane contains information and editable attributes about currently selected items. In the **Corner Tool**, selecting the junction selects all four corners of the junction, so you can modify the attributes of all four corners at the same time.

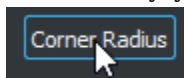
In the **Attributes** pane, set the **Corner Radius** attribute of all four corners to 10.



The junction corners expand in the scene editing canvas.



Alternatively, you can modify the **Corner Radius** attribute value by clicking on the attribute name

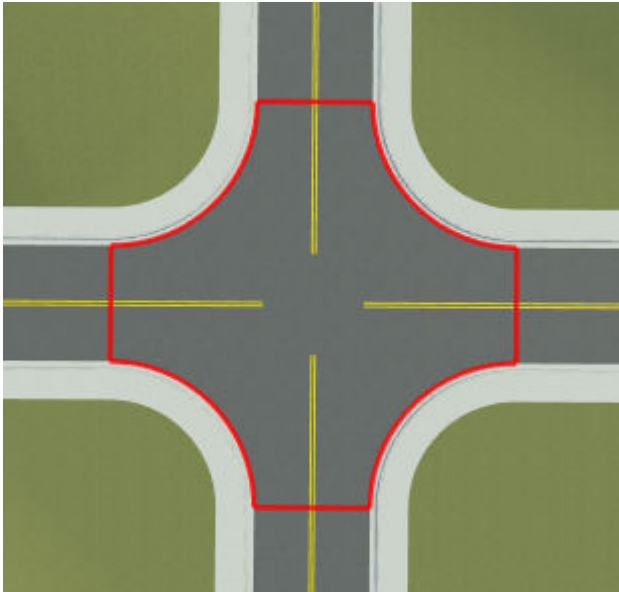


and dragging up or down.

Add Crosswalk

Add a crosswalk to the intersection.

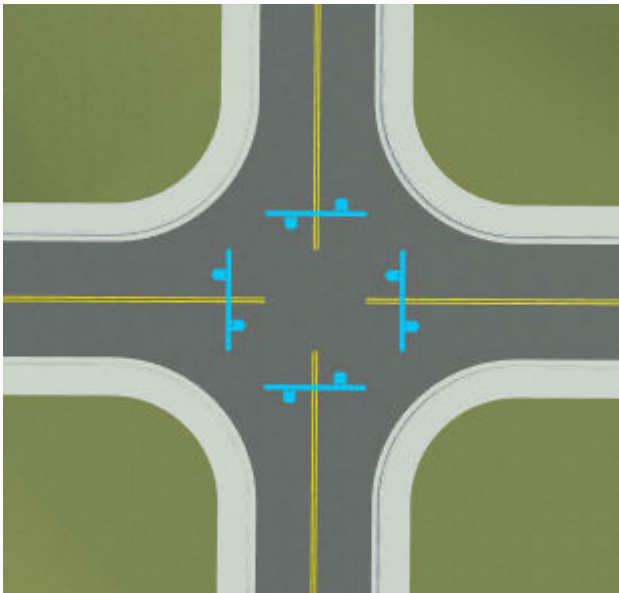
- 1 Rotate the camera to view the intersection from the top down. To focus the camera on the selected intersection, press the **F** key.



2

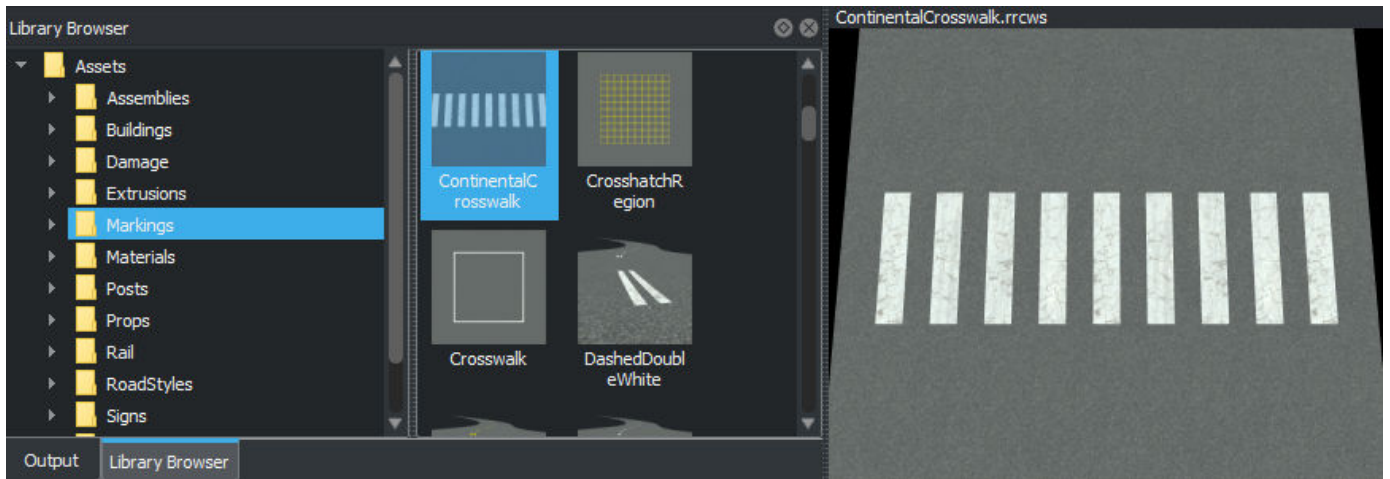


Click the **Crosswalk and Stop Line Tool** button. The intersection displays blue chevrons for adding stop lines to the intersection.

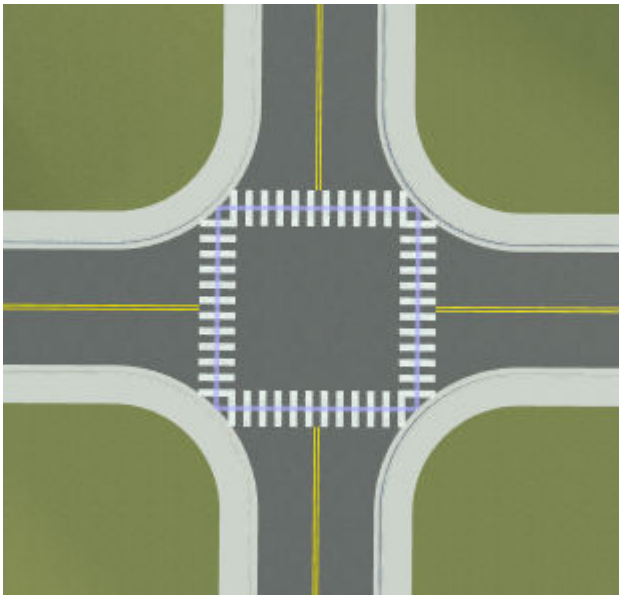


3 From the **Library Browser**, select a crosswalk to add to the intersection. The **Library Browser** stores all assets available to add to a scene. Assets include 3D objects, markings, textures, and materials.

In the **Library Browser**, select the **Markings** folder, and then select the **ContinentalCrosswalk** asset. A preview of the asset displays in the asset viewer.



- 4 Click within the intersection to clear the blue chevrons. Then, right-click in the intersection to apply the selected crosswalk asset to the intersection.



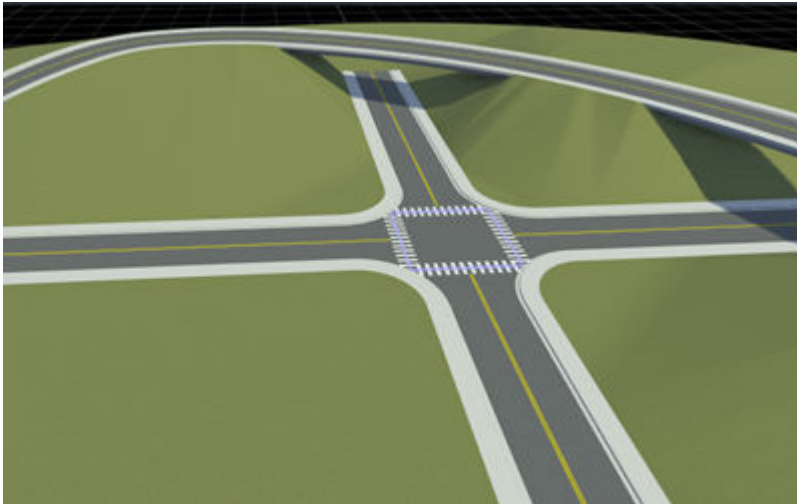
Add Turning Lanes

Convert one of the roads at the intersection into a more complex highway road that includes a turning lane with arrow markings.

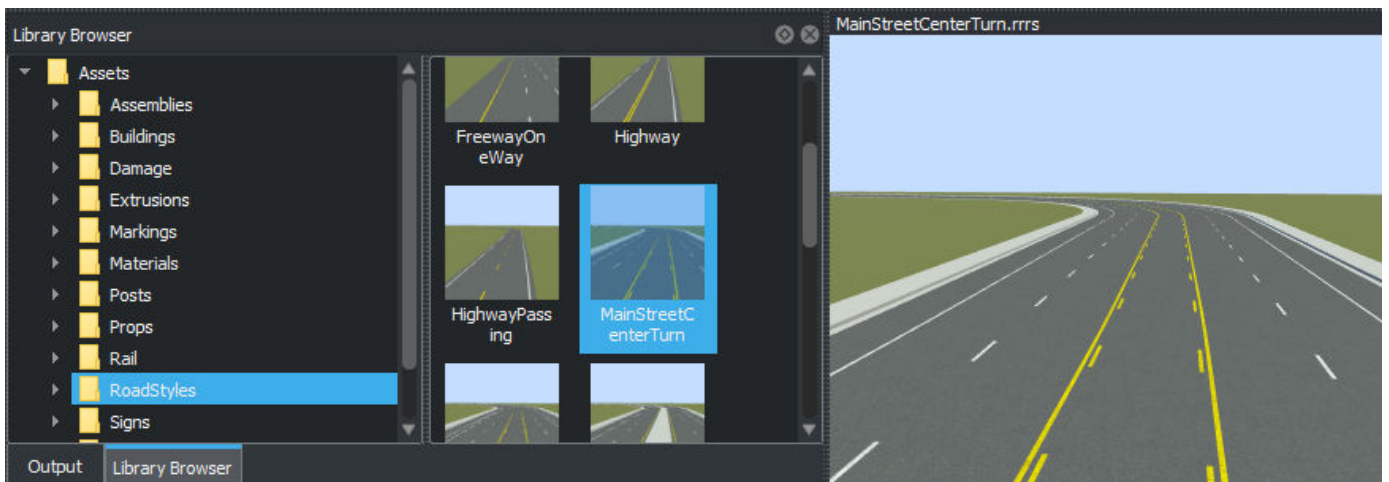
Change Road Style

The existing roads all use the default road style, which is of a simple two-lane divided highway with sidewalks. Update one of the roads at the intersection to use a road style with additional lanes.

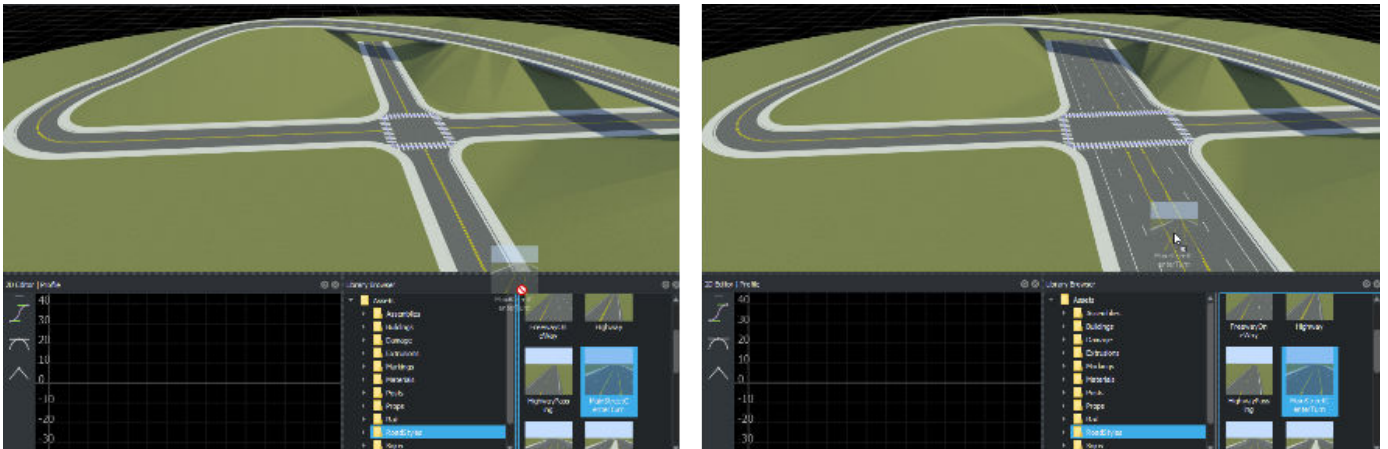
- 1 Zoom out and rotate the camera to view the scene at an angle similar to the one shown here.



- 2 In the **Library Browser**, open the RoadStyles folder, and then select the **MainStreetCenterTurn** asset. This road style asset includes shoulder lanes, two passing lanes on each side, and a median lane. Optionally, rotate and move the camera in the asset viewer to inspect the road style.



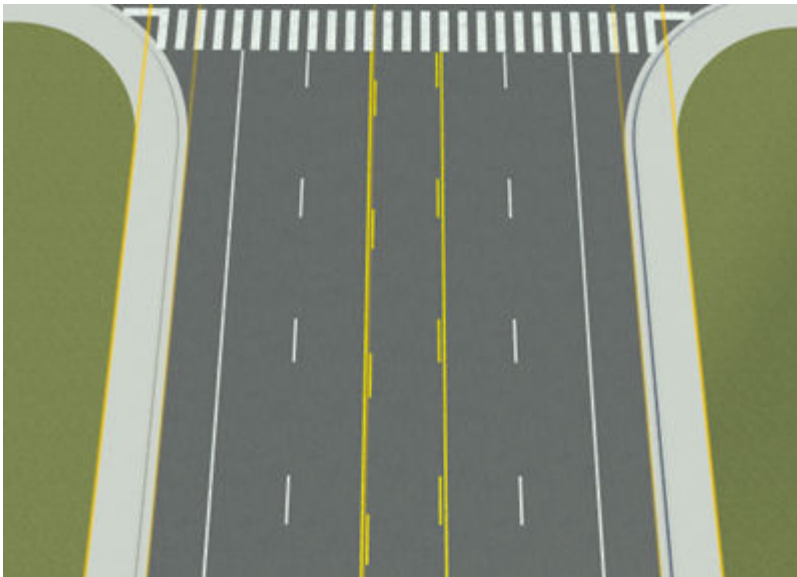
- 3 Drag the selected road style onto the road closest to the camera, as shown here. The road updates to the new style and switches back to the **Road Plan Tool**. The road maintains the corner radius and crosswalk style previously applied.



Create Turning Lane at Intersection


Create a short left-hand turn lane near the intersection.

- 1 Rotate the camera and zoom in near the crosswalk on one side of the road that has the new road style.

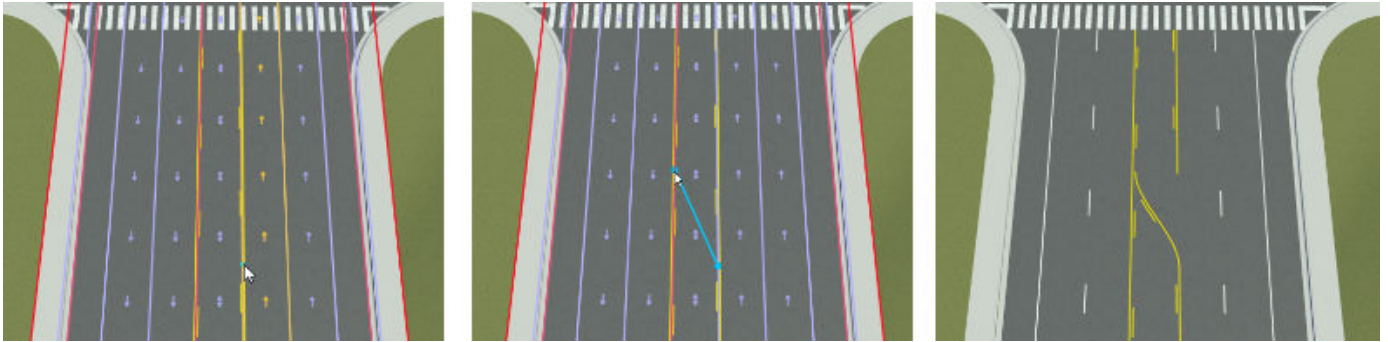


2

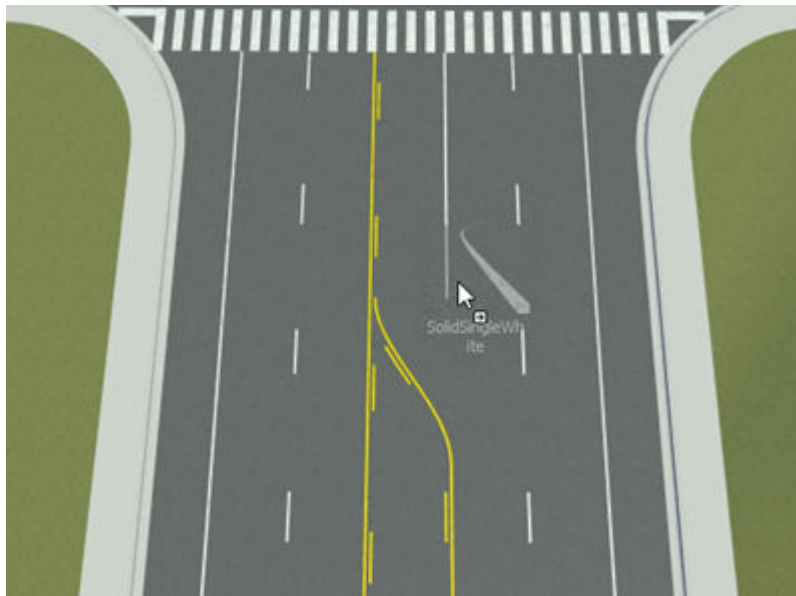


Click the **Lane Curve Tool** button . This tool enables you to create a tapering cut in an existing lane to form a turning lane.

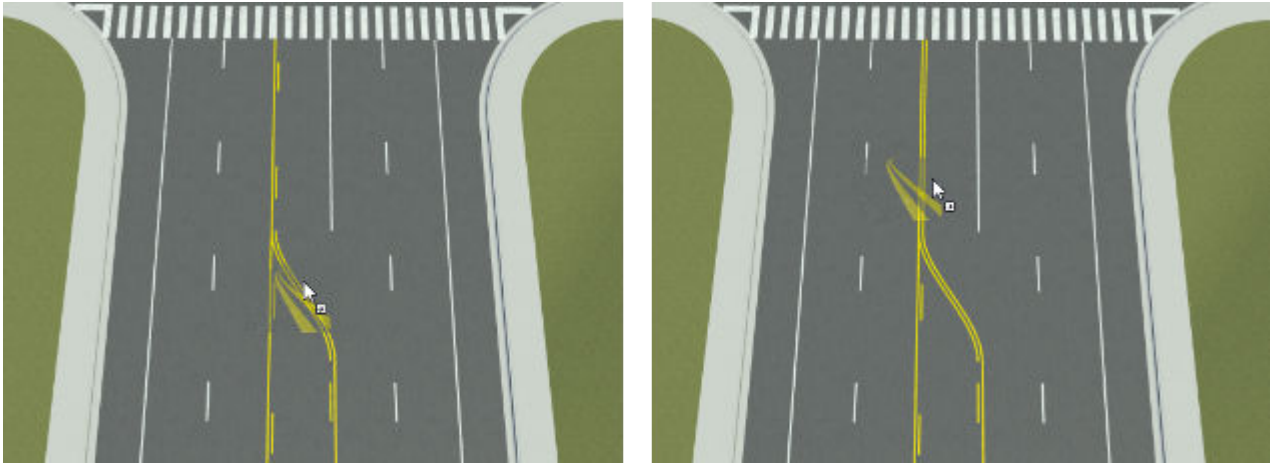
- 3 Click to select the road. Then, right-click the right side of the median lane where you want the tapering cut to start. Drag the blue line diagonally to the left side of the median lane where you want the tapering cut to end and the turning lane to start.



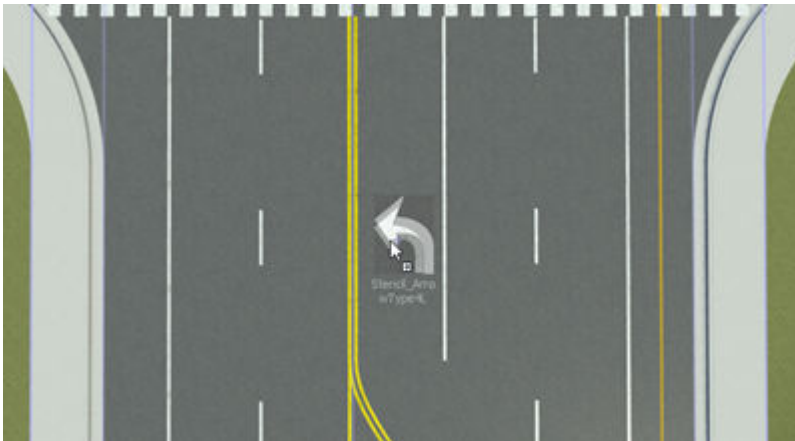
- 4 The newly formed turning lane still has the styles of the median lane. Update the lane markings to match the style of a standard turning lane.
 - a In the **Library Browser**, select the `SolidSingleWhite` asset and drag it onto the right side of the turning lane. The lane marking changes to a solid single white line.




- b Select the `SolidDoubleYellow` asset and drag it onto the two marking segments that form the left side of the turning lane. The lane marking segments change to solid double yellow lines.

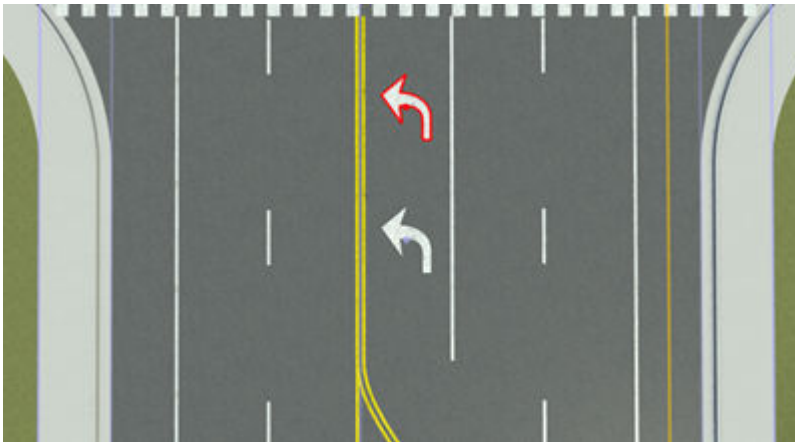


- 5 Add a turning arrow to the lane. In the **Stencils** folder of the **Library Browser**, select the **Stencil_ArrowType4L** asset. Drag this asset into the turning lane at the point where you want to add the arrow stencil.

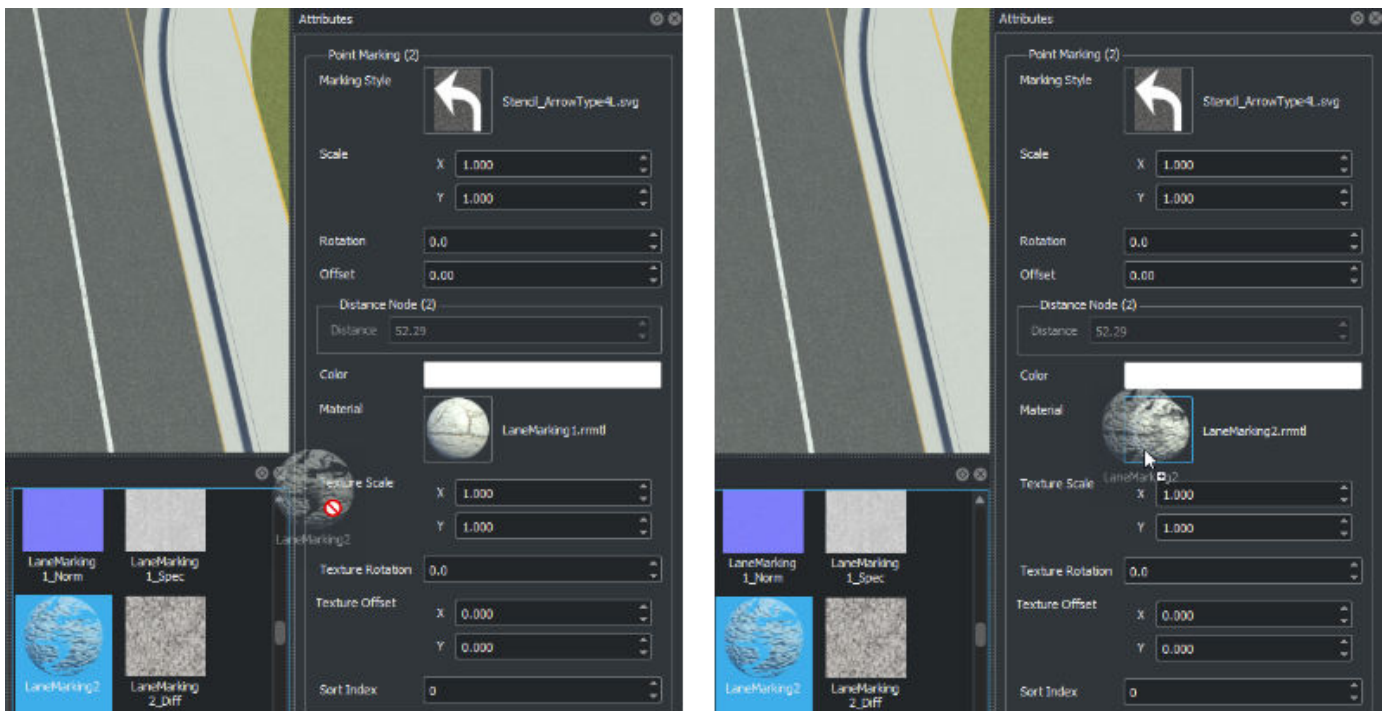


6

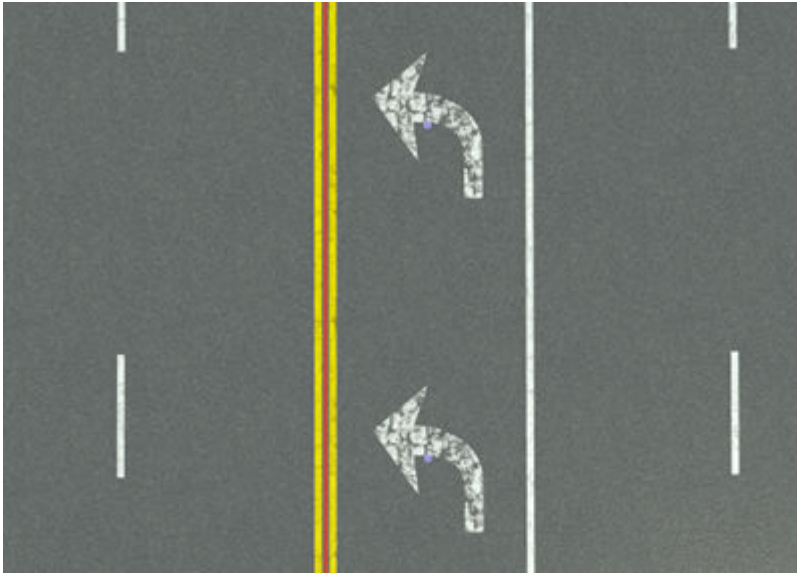
By adding the arrow stencil, RoadRunner selects the **Marking Point Tool**  to make it the active tool. You can now add the second arrow by right-clicking at the point where you want to add it.



- 7 Modify the marking material of the arrows to make them appear more worn. First, select the two arrows. In the Markings folder of the **Library Browser**, select the LaneMarking2 material asset. Then, drag this asset into the **Attributes** pane for the selected arrows and over the existing LaneMarking1 material asset.



The arrows update to use the new more worn-looking material.



Repeat these steps to create the turning lane on the other side of the intersection.



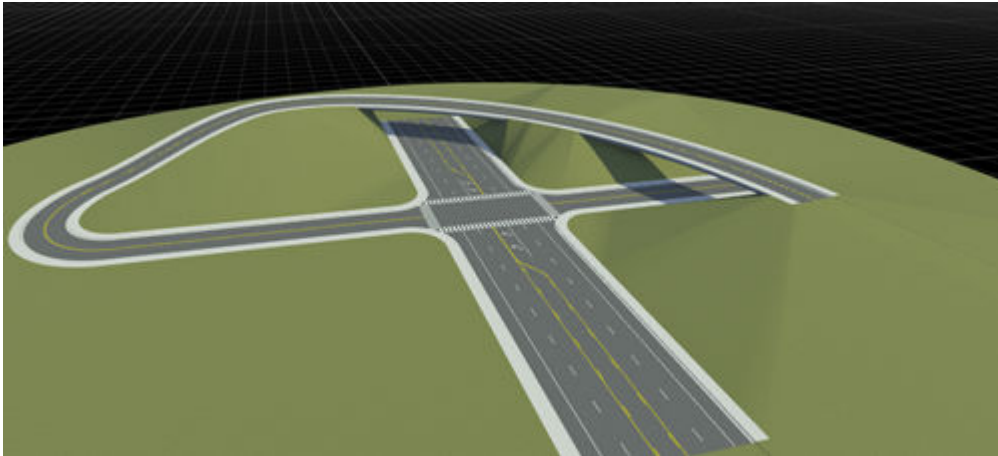
Add Props

To enhance the scene with more detail, add props to it. Props are 3D objects such as posts, poles, and signs that you can place on and around roads. Add tree props around the road using multiple techniques.


Add Individual Props

Add bushes to one section of the terrain.

- 1 Zoom out and rotate the camera to fit the entire road network and surrounding terrain in view.




- 2 In the **Library Browser**, open the Props folder and select the Trees subfolder.
- 3 Select a bush prop (one of the asset files that begins with Bush_). Drag the bush onto a section

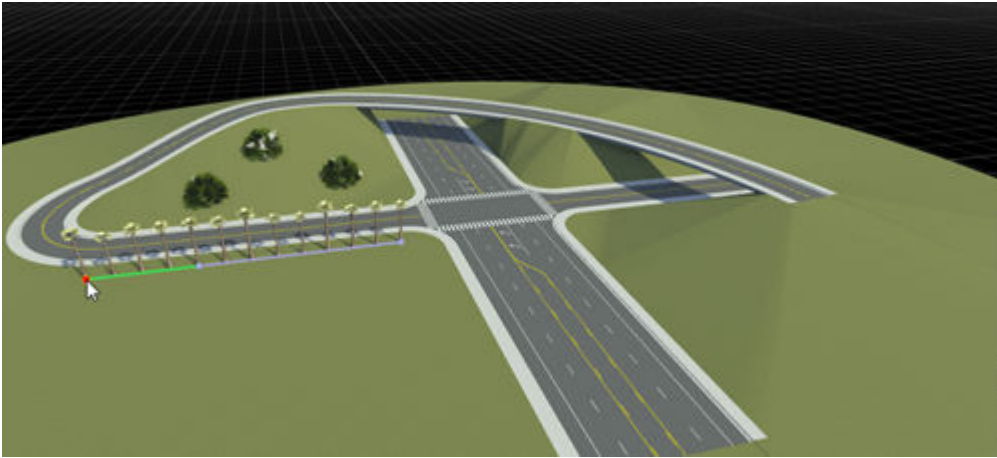
of the scene. RoadRunner switches to the **Prop Point Tool** . Drag additional bushes into the scene or right-click to add more bushes. All the bushes are aligned with the surface terrain.



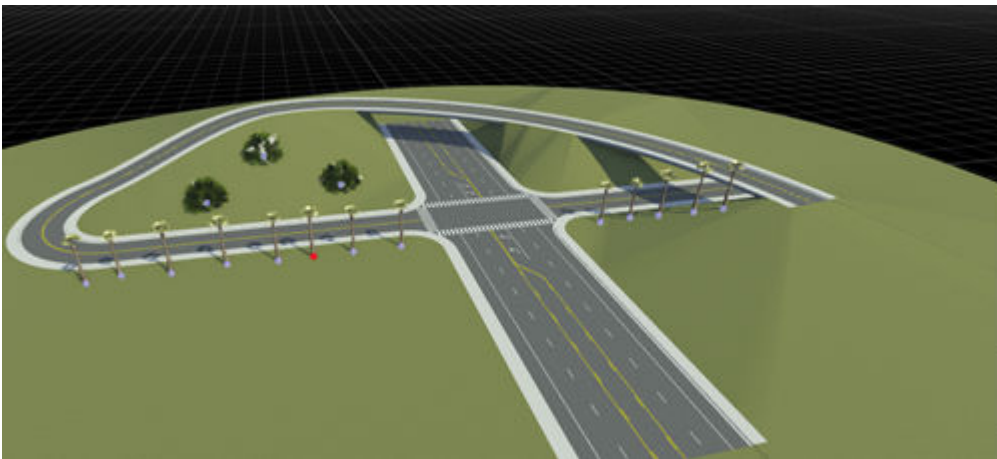
Add Props Along Curve


Add props along a curve to follow the edge of the road.

- 1 Click the **Prop Curve Tool** button .
- 2 In the **Library Browser**, in the Trees folder, select a California palm tree prop (one of the asset files that begins with CalPalm_).
- 3 Right-click along the road edge of one side of the intersection to add a line of palm trees to it. Click away from the prop curve to complete the line.



- 4 To make each individual tree in the span moveable and selectable, you can convert the curve to individual props. Select the prop curve, and in the **Attributes** pane, click **Bake**. The palm trees become individual props and RoadRunner switches to the **Prop Point Tool**. Move some of the palm trees to the other side of the intersection.



Alternatively, to add a prop along a span of road, you can click the **Prop Span Tool** button , select a road, and drag the prop onto the road edge.

Add Props in Specified Area

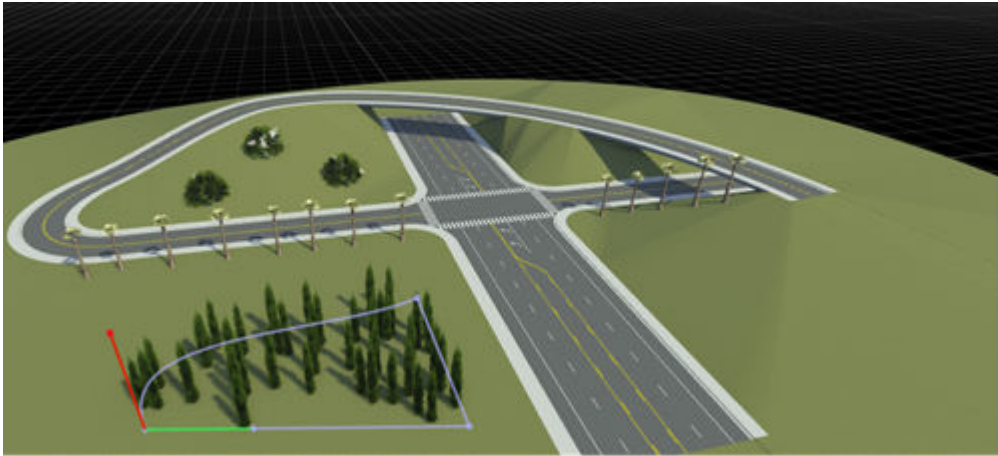
Add props in a specified area of the ground surface.

1

Click the **Prop Polygon Tool** button .

2 In the **Library Browser**, in the Trees folder, select a cypress tree prop (one of the asset files that begins with Cypress_).

3 Right-click within an empty area of the surface terrain to draw a polygon containing the selected prop. Click away from the polygon to finish drawing it. Then move the points or tangents to change the shape of the polygon.

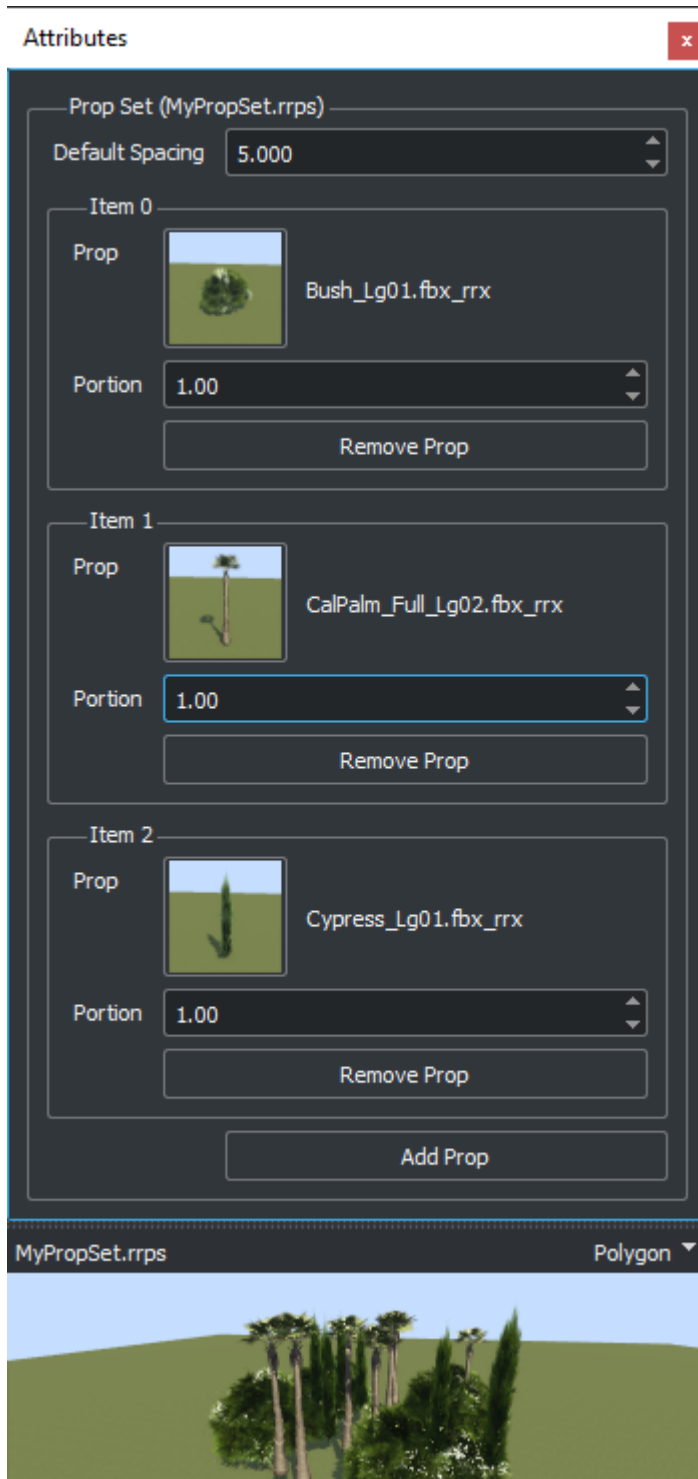


- 4 Optionally, modify the prop polygon by using the attributes in the **Attributes** pane. For example, to increase or decrease the number of props in the polygon, use the **Density** attribute. To randomize the distribution of assets in the polygon, click **Randomize**.

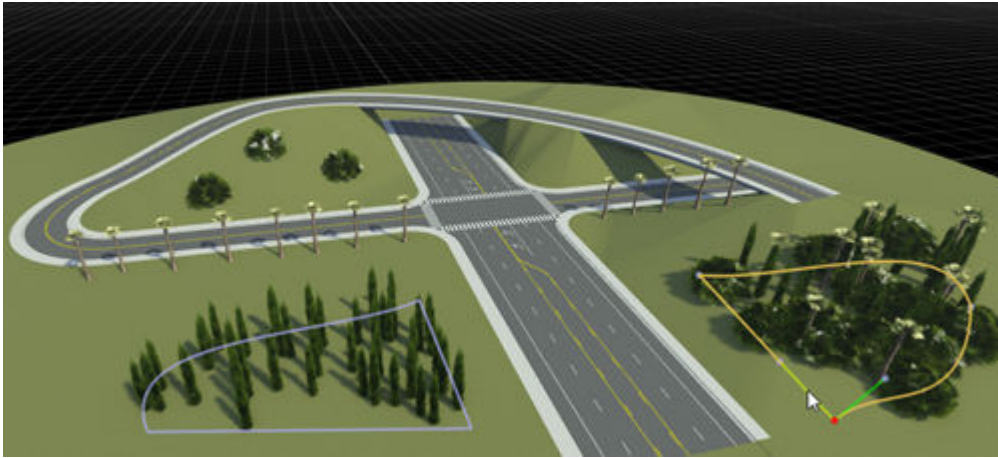
Add Props of Varying Types

So far you have added a single type of prop to the scene. To add a variety of props to a scene simultaneously, you can create a prop set.

- 1 In the **Library Browser**, in the **Trees** folder, hold **Ctrl** and select the three props you added to the scene in the previous sections.
- 2 Select **New**, then **Prop Set** and give the prop set a name. The new prop set is stored in the **Trees** folder. The **Attributes** pane displays the three props in the set and a preview of the prop set.



- 3 Click the **Prop Polygon Tool** button. Create a prop polygon on an empty part of the terrain that contains the new prop set.



Optionally, you can also replace the existing cypress tree props with the new prop set by dragging the prop set onto the polygon of cypress trees.

Other Things to Try

You have now created a simple road network containing a realistic turning lane, multiple overpasses, and trees of varying types.



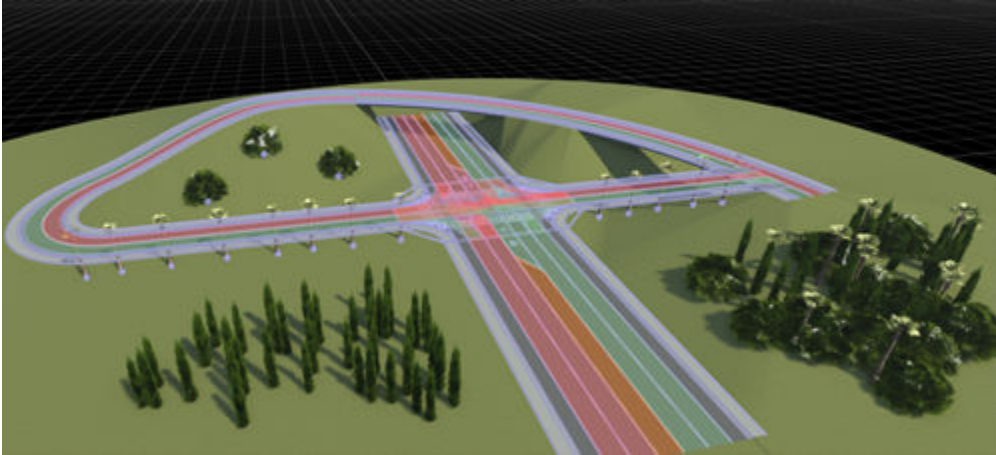
You can now enhance the scene using additional tools. For example, try these things:

- Add more roads or connect the existing roads in the scene. To smooth the transitions between roads that have different numbers of lanes, use lane tools such as the **Lane Tool**, **Lane Width Tool**, **Lane Add Tool**, or **Lane Form Tool**.
- Add traffic signals to the intersection by using the **Signal Tool**. To modify the paths through the lanes at each turn signal, use the **Maneuver Tool**.
- Add additional props to the scene, such as barrels, buildings, and traffic signs. To modify the text of signs, use the **Sign Tool**.

In addition, you can try exporting the scene to one of the supported export formats. These export options are on the **File** menu, under **Export**. To customize export options before exporting, use the **Scene Export Preview Tool**. If you are exporting to OpenDRIVE, use the **OpenDRIVE Export**

Preview Tool. This image shows how the export preview of the scene you created looks when you

click the **OpenDRIVE Export Preview Tool** button .



If you want to create a new scene that is based on a real-world location, then you can import geographic information system (GIS) data such as aerial imagery into RoadRunner and create scenes around it. For an example, see “Create Roads Around Imported GIS Assets” on page 1-55.

See Also

Related Examples

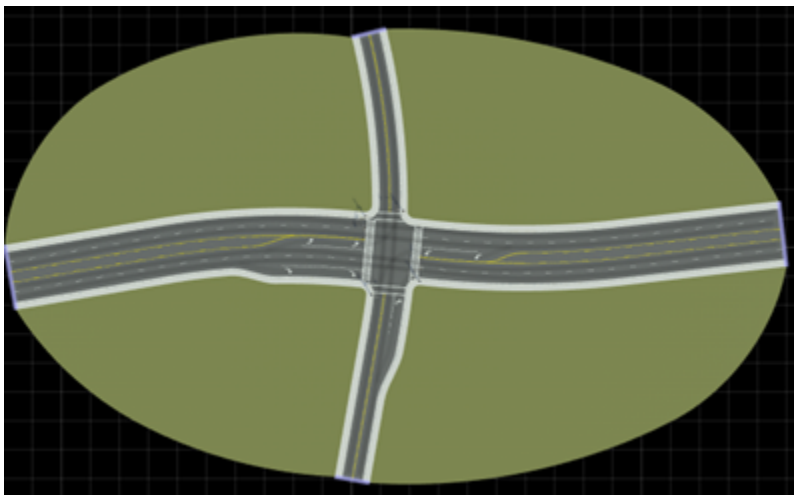
- “Window Layouts” on page 2-6
- “Camera Control in RoadRunner” on page 1-44
- “Create Roads Around Imported GIS Assets” on page 1-55
- “Choose a RoadRunner Tool” on page 2-27
- “Keyboard Shortcuts and Mouse Actions for RoadRunner” on page 2-23

Camera Control in RoadRunner

RoadRunner enables you to edit large-scale and small-scale details of a 3D environment that can span many kilometers or miles. The interactive camera controls enable you to navigate this large 3D space quickly and effectively. This example shows you the fundamentals of camera controls in the RoadRunner scene editing environment.

Open Scene

Open a basic scene to move the camera around in. From the menu bar, select **File**, then **Open Scene**. Then, open `FourWaySignal.rrscene`, which is one of the default scenes that is included in the `Scenes` folder of RoadRunner projects.



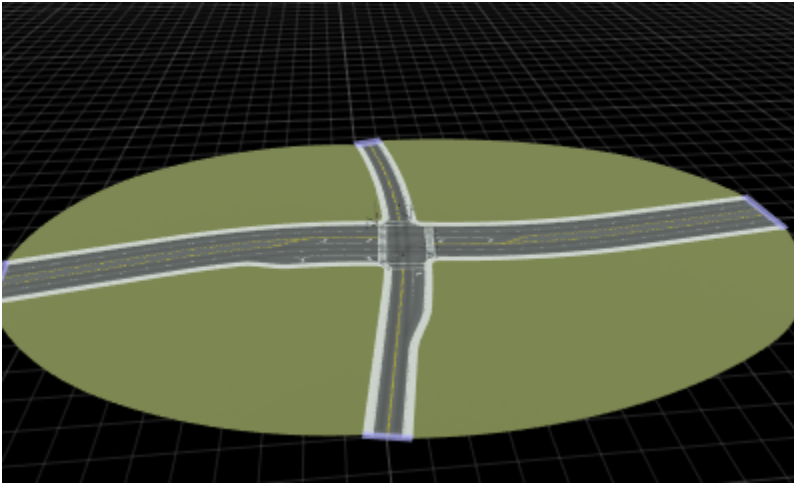
Rotate Camera

Camera control in RoadRunner is based on a polar viewing model, where the camera orbits around a point of interest at a fixed distance. By default, when you open a new scene, the point of interest is 1.5 meters above the origin, to approximate the position of the head of a person standing at the center of your scene. The point of interest for this scene is at the center of the intersection.

You can rotate the camera around the point of interest at any time and from within any tool by pressing and holding either the **Alt** key or the **Windows** key and moving the pointer.

Note In Linux, Ubuntu 16.04, pressing the **Alt** key moves the current window and pressing the **Windows** key shows certain help overlays. To use the **Windows** key instead of the **Alt** key for moving windows, update Ubuntu 16.04 according to the instructions in “Update Linux Ubuntu Key Mapping” on page 2-25.

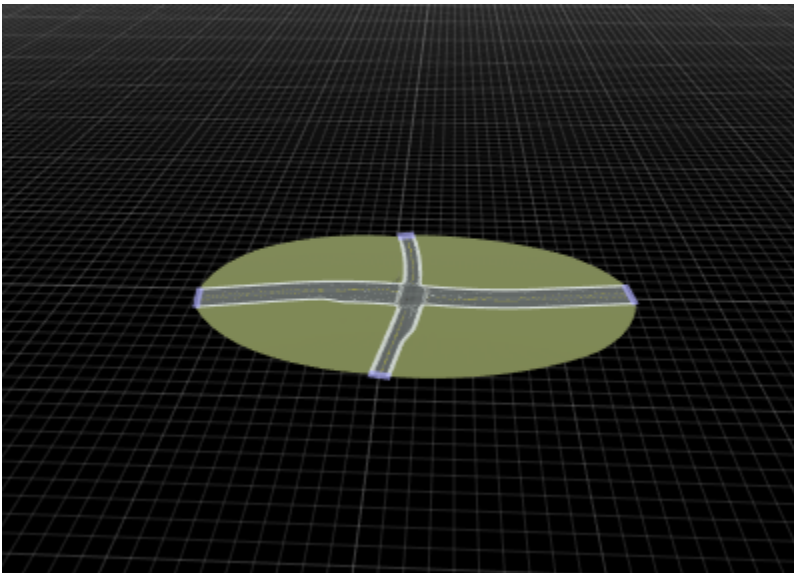
Hold the **Alt** key, click anywhere in the scene, and drag the pointer. Observe the change in camera rotation with respect to the point of interest.



Zoom Camera In and Out

To zoom the camera in, hold **Alt** and the right-click button, and then drag the pointer up or right. Conversely, to zoom the camera out, hold **Alt** and the right-click button, and then drag the pointer down or left. Alternatively, you can use the mouse scroll wheel to zoom in or out.

Hold **Alt** and the right-click button, and then drag the pointer down to zoom out. Observe the change in the fixed distance at which the camera orbits the scene.



While still holding **Alt** and the right-click button, drag the pointer up and zoom all the way in to the point of interest until the camera stops moving. Zooming in this far focuses the camera at the pavement.



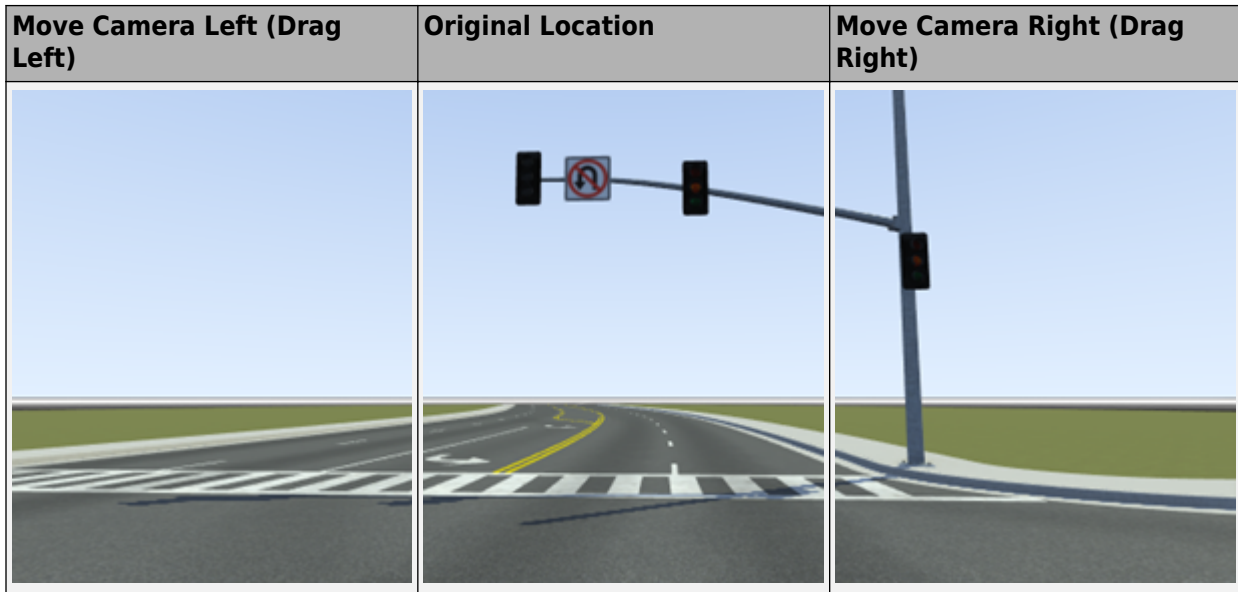
Hold **Alt** and the left-click button to rotate the camera. The camera movement at this distance is similar to standing at a fixed location and looking around as the camera orbits the scene.



Move Camera Horizontally

To move the camera horizontally along the ground (xy) plane, first hold the **Alt** key and the left-click and right-click buttons. Then, drag the pointer. Alternatively, you can move the camera by holding the middle-click button and dragging the pointer.

Hold **Alt**, the left-click button, and the right-click button, and then drag the pointer to move the camera to the left and right. The point of interest shifts to the new location.



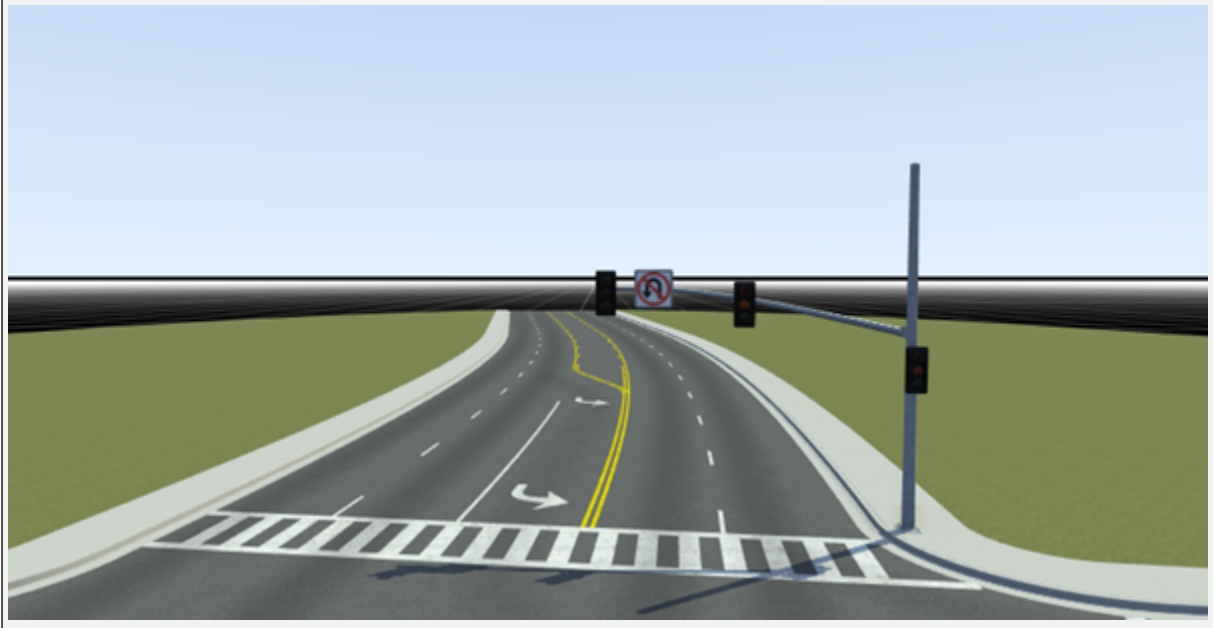
Move Camera Vertically

For simple environments, you can keep the height of the camera point of interest set to the default. However, for more complex environments, you might need to move the point of interest up or down. For example, if you are designing a scene with bridges, you might need to move the point of interest down so that you can maneuver the camera under a bridge.

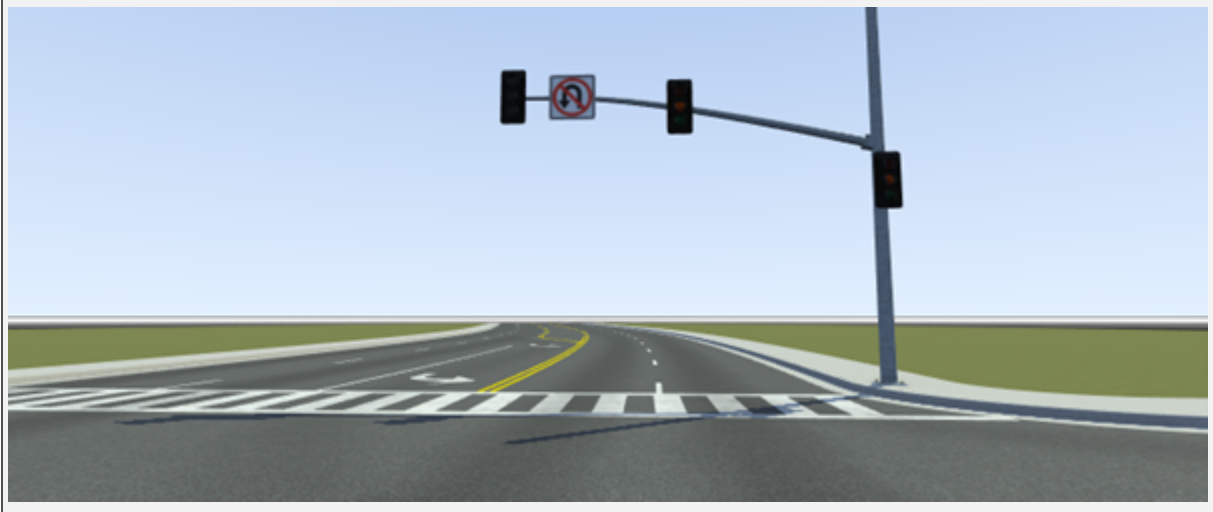
To move the camera up, hold **Alt**, **Shift**, left-click, and right-click, and then drag the pointer down. Conversely, to move the camera down, hold these same keys and buttons, and then drag the pointer down. Alternatively, you can hold **Alt**, **Shift**, and the middle-click button and then drag the pointer up or down for the same affect.

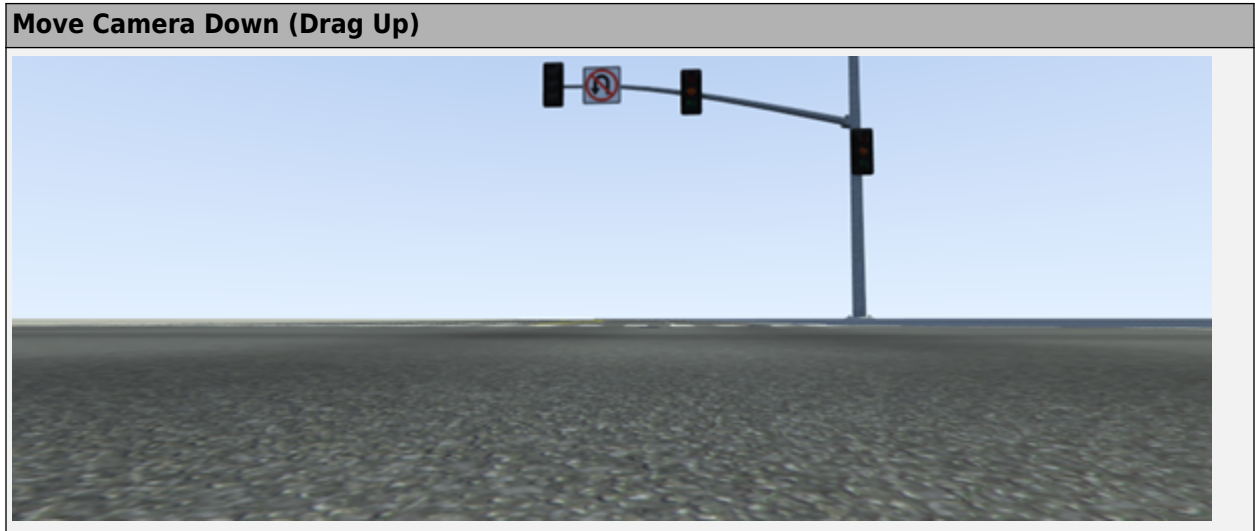
Hold **Alt**, **Shift**, left-click and right-click, and then drag the pointer up and down. Observe the change in the point of interest as the camera moves up and down.

Move Camera Up (Drag Down)



Original Location

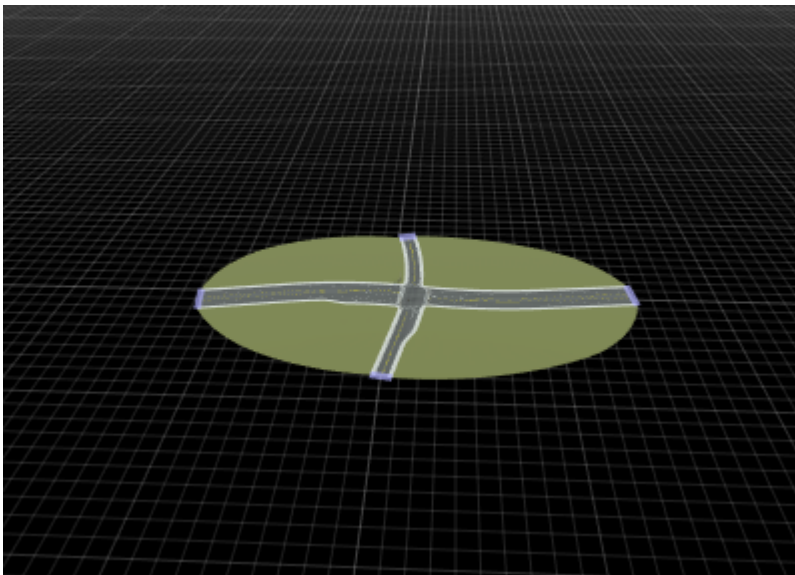




Frame Camera on Selected Object

From within any tool, you can center, or frame, the camera on the currently selected objects. To center the camera on a selected object, press the **F** key. Alternatively, from the **View** menu, select **Frame Selected**.

- 1 Zoom out of the scene. Hold **Alt** and the right-click button, and then drag the pointer down.

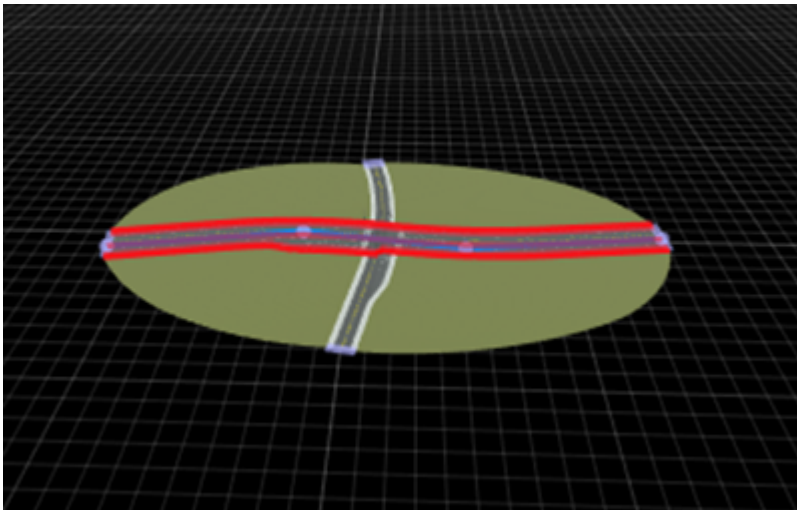


2

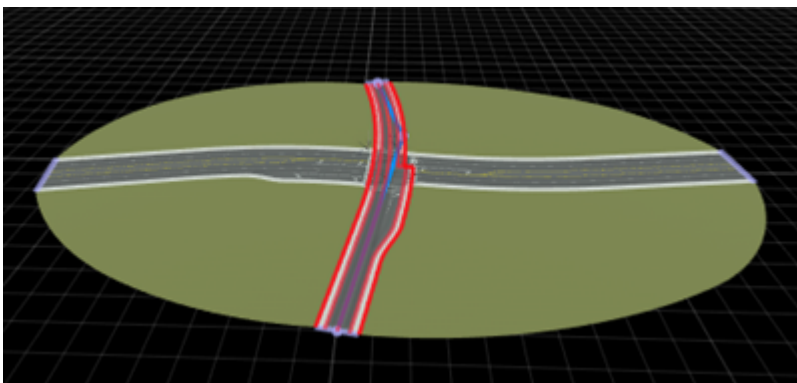


Click the **Road Plan Tool** button to make the roads selectable.

- 3 Select the longer road and press the **F** key. The camera centers on the longer road.



- 4** Select the other road and press **F**. The camera zooms in to center on the shorter road.

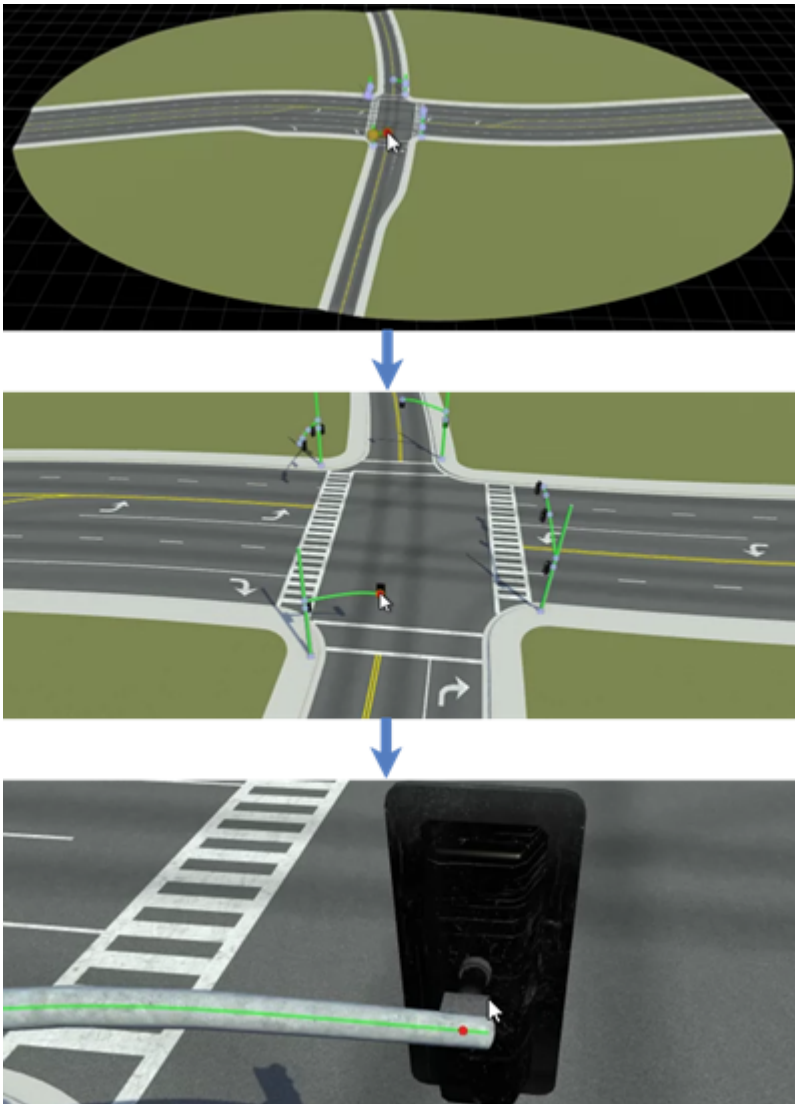


- 5**

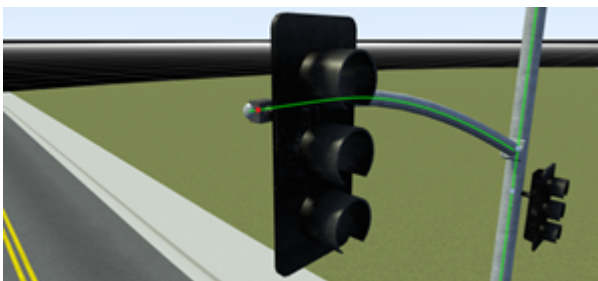


Click the **Prop Point Tool** button to make the traffic light props selectable.

- 6** Select one of the prop points and press **F**. The camera zooms in on the selected prop.



When you rotate the camera by holding **Alt** and then clicking and dragging, the camera rotates around the prop.



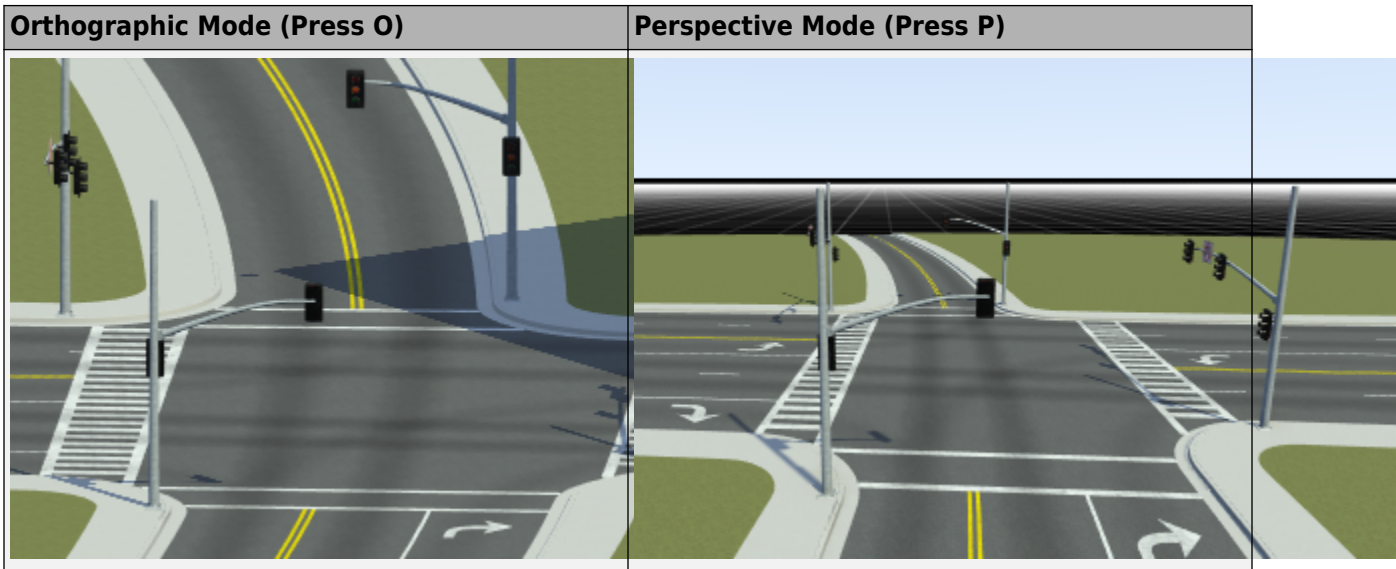
Change View Projections

The RoadRunner camera can use either a perspective or orthographic viewing projection.

- The perspective projection is the default viewing projection, which causes distant objects to appear smaller than close objects.
- The orthographic projection is similar to what you might find in a CAD tool. It is useful for precise positioning, usually from a top-down point of view. In orthographic mode, objects do not change apparent size as they get closer or further away.

The camera controls work the same in both projection modes.

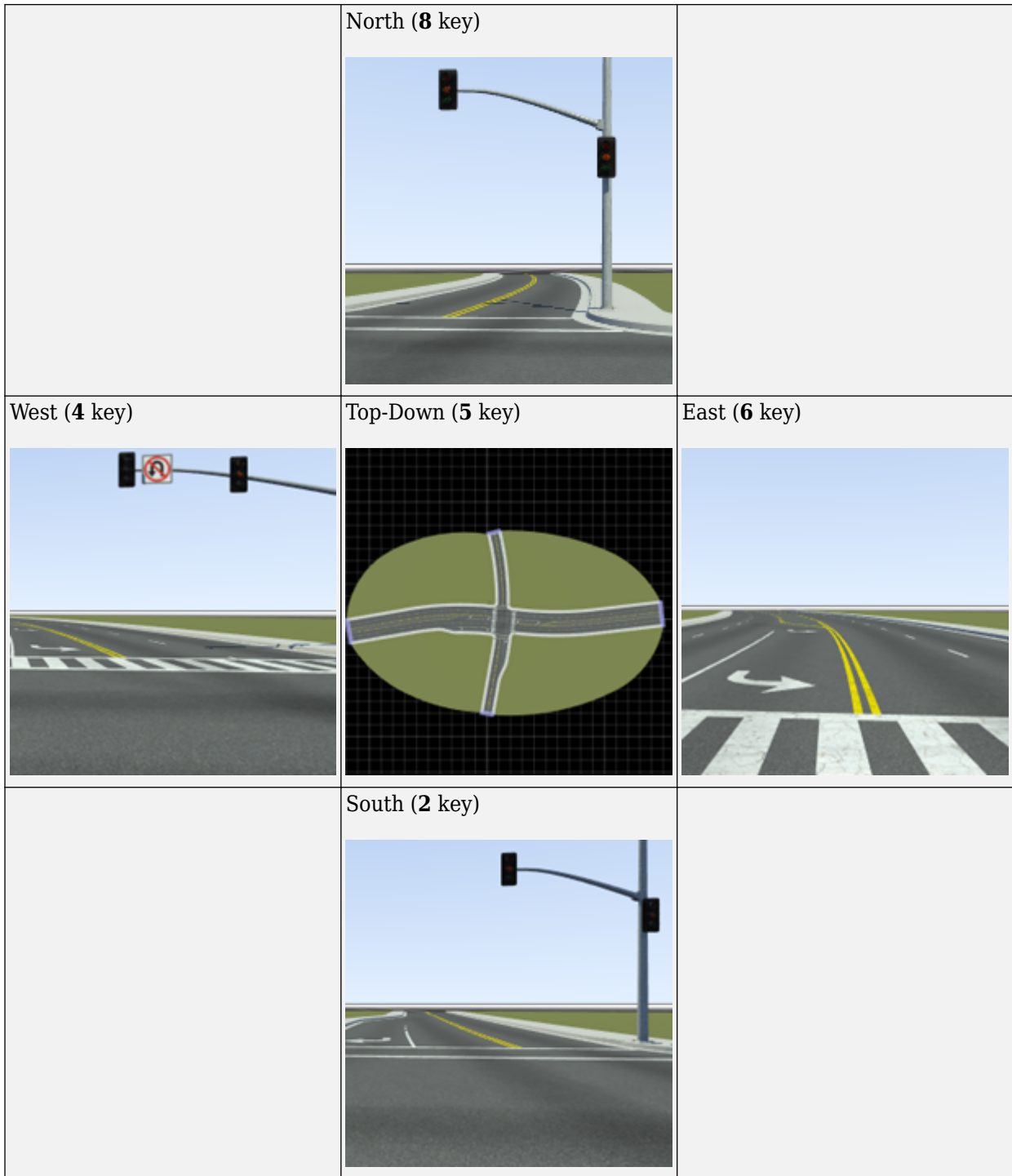
Move the camera so that the entire intersection is in view. Then, press **O** to switch to orthographic mode. In this mode, the traffic lights are all the same size. Press **P** to switch back to perspective mode, where the traffic lights in the distance appear smaller.



Set View Direction of Camera

You can set the view direction of the camera to due north, south, east, west, or top-down. To change the view direction, on the **View** menu, select **Direction**, and then select the view direction you want. Alternatively, you can use number pad shortcut keys.

Change the view direction of the scene by using these keys on the number pad. This table shows sample view directions when the camera is at the intersection of the scene and their corresponding number pad shortcut keys. In the top-down view, the camera also rotates to point north.



See Also

Related Examples

- "Create Simple RoadRunner Scene" on page 1-19

- “Keyboard Shortcuts and Mouse Actions for RoadRunner” on page 2-23

Create Roads Around Imported GIS Assets

RoadRunner supports a variety of geographic information system (GIS) formats. You can import GIS assets into RoadRunner and use them as a reference to construct your road network.

This example shows how to create roads around imported GIS assets with RoadRunner for an area in Downey, Los Angeles, California, United States. This example uses GIS data obtained from the Basic National Map explorer interface.

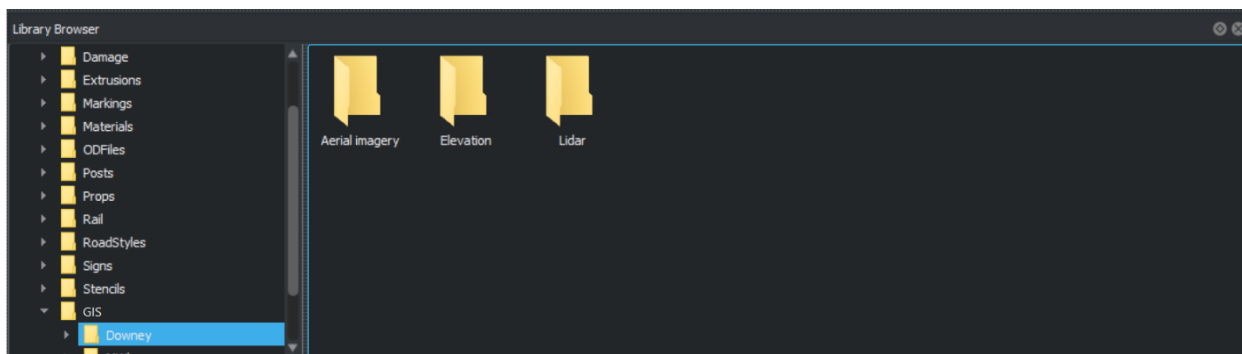
The example uses three different types of GIS assets:

| GIS Asset | Description |
|----------------|---|
| Aerial Imagery | Visual reference for roads and surface texture mapping |
| Point Cloud | Visual reference and object placement information (trees, buildings, markings, and so on) |
| Elevation | Height information of the terrain |

Download and Import GIS Assets into RoadRunner

- 1 Download aerial imagery, elevation, and point cloud data of a specific location. To learn how to download GIS data supported by RoadRunner, see “GIS Data Resources for RoadRunner” on page 3-7.
- 2 Open a RoadRunner project and create a new scene by selecting **File**, then **New Scene** from the main menu.
- 3 To store the imported assets in the **Library Browser**, right-click in the **Library Browser** and select **New**, then **Folder**. Alternatively, from the main menu, select **Assets**, then **New**, then **Folder**, and name the folder.
- 4 Optionally create subfolders to organize the assets. Move the downloaded assets to the respective folders by dragging and dropping the aerial imagery, elevation, and point cloud files into the **Library Browser**.

In this **Library Browser** image, the assets are stored in Aerial Imagery, Elevation, and Lidar folders. These folders are contained in a Downey subfolder under a top-level GIS folder.



For more details on importing assets, see “Create, Import, and Modify Assets” on page 2-42.

Set World Origin

- 1 To import the GIS asset around a specific geographic position, use the **World Settings Tool**



. For this example, the world origin latitude is set to 33.9383 degrees, and the world origin longitude is set to -118.1296 degrees. Click **Apply World Changes**. The region displayed on the RoadRunner canvas defines the area where you can load the GIS asset.

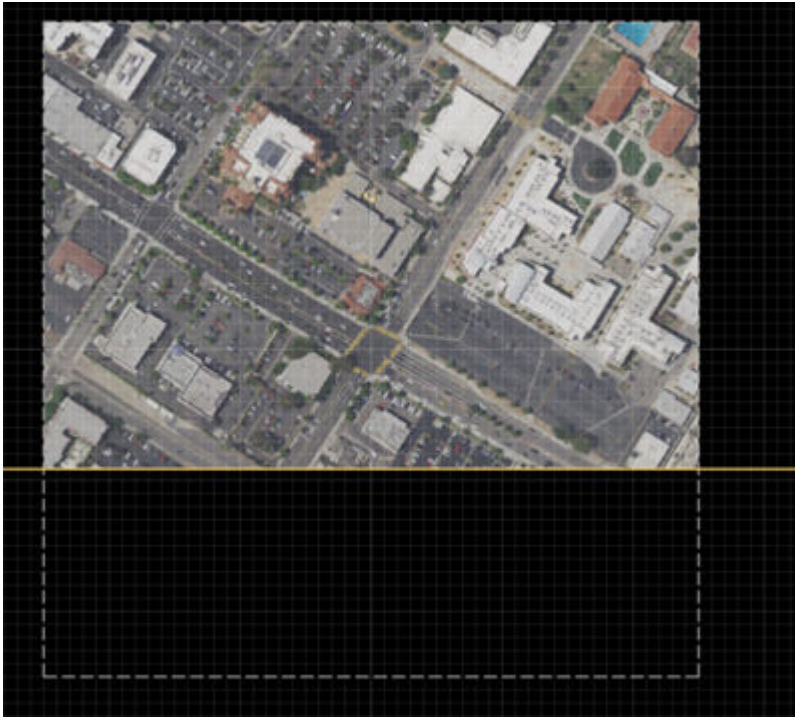


- 2 Optionally change the size of the region by using the **Extents** attribute with the **World Settings Tool** selected.

Add GIS Assets

After you set the origin, import the aerial imagery, elevation, and point cloud data into the scene.

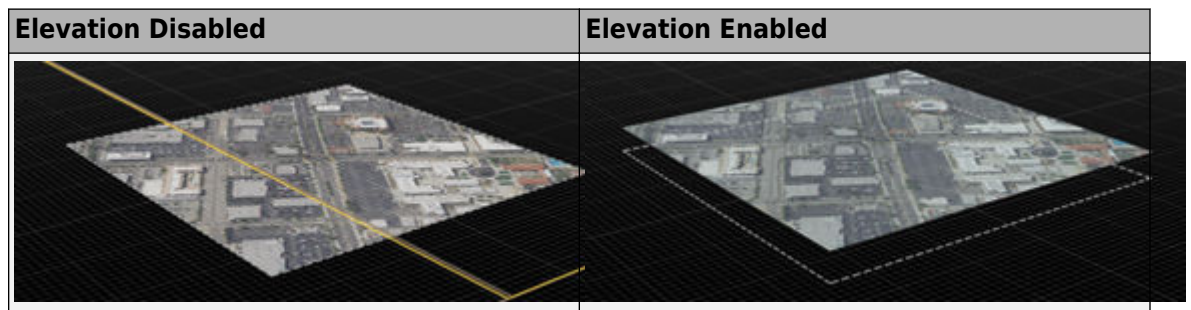
- 1 In the **Library Browser**, navigate to and select the aerial imagery file, making sure that it has a projection. To check the projection of an asset, click the file and in the **Attributes** pane, check the **File Projection (WKT)** section.
- 2 Drag the GIS asset file into the scene. While loading the GIS asset, a rotating orange wheel on the top right of the RoadRunner canvas appears, indicating the loading progress of the file. RoadRunner places the file in the correct world and scene position with respect to the specified **World Origin**. For example, in this image, the aerial imagery dataset covers the top portion of the scene.



- 3 Similarly, to add the elevation files, drag the files into the scene. You can drag multiple files of the same type into the RoadRunner scene editing canvas.

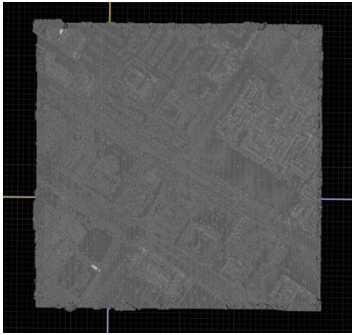
Dragging elevation files into a scene automatically switches RoadRunner to the **Elevation Map Tool** and adds the appropriate elevation to the scene, projected with respect to the world origin.

To toggle the visibility of the elevation data, from the **View** menu option, select **Elevation Map (F5)**. RoadRunner provides additional options for toggling the visibility of other GIS asset types.

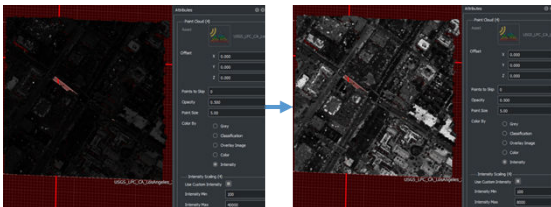


- 4 Similarly, to add point cloud data, drag the files into the scene. Check that it has a projection listed in the **File Projection (WKT)** attribute. You can select multiple files and drag them into the scene. To toggle the visibility of the point cloud data, from the View menu option, select **Point Cloud (F6)**.

The point cloud data, as shown by the grey dots, are automatically placed in the right regions with respect to the world origin.




- 5 Customize the point cloud appearance and feature visualization. With the **Point Cloud Tool** still enabled, in the **Attributes** pane, set **Color By** to **Intensity**. Then, enable **Use Custom Intensity** and adjust the **Intensity Min** and **Intensity Max** values to make the scene more visible. This image shows the difference in intensity when the maximum intensity is reduced from 40000 to 8000.

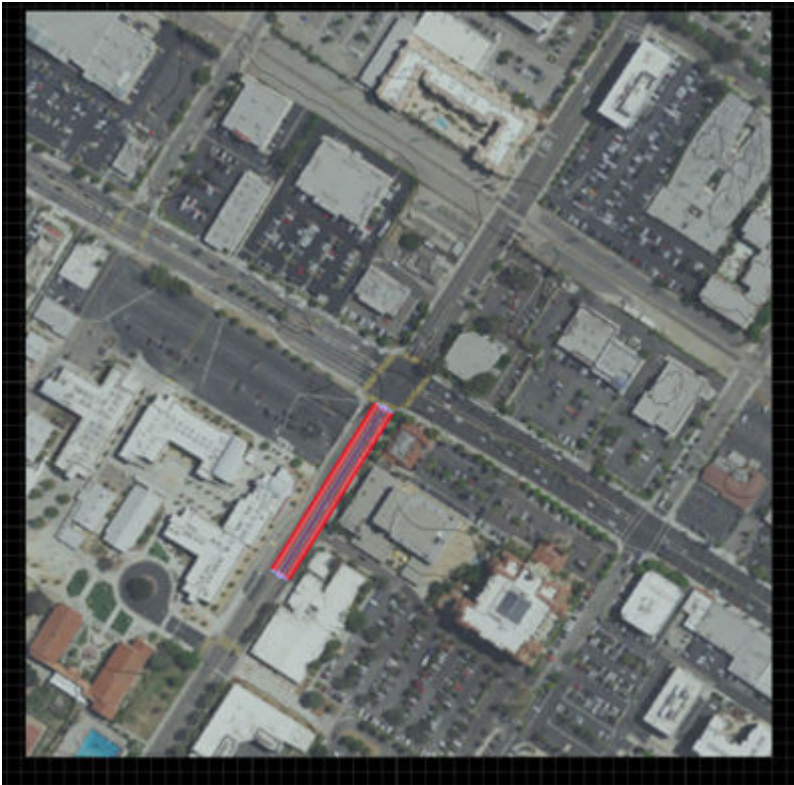



Create Roads Around GIS Assets

After importing the GIS assets, use various tools within RoadRunner to customize the scene.

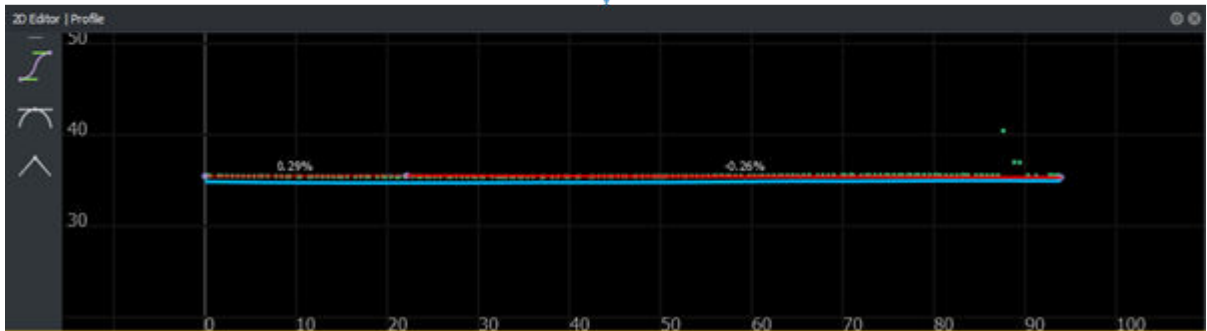
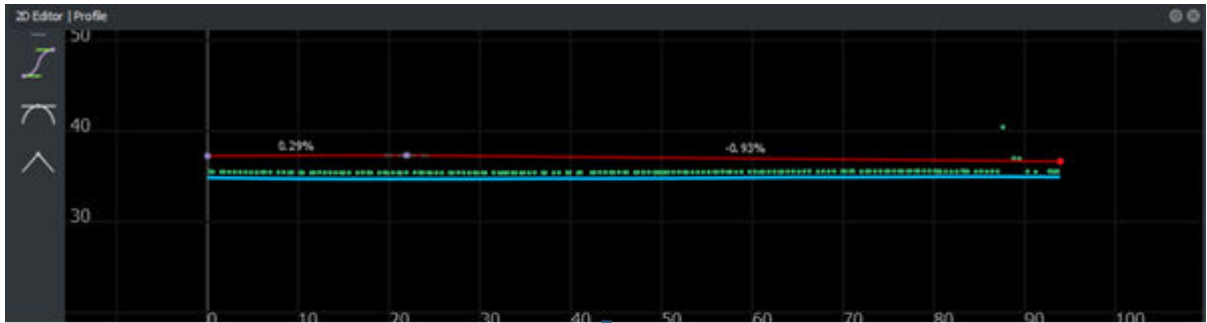
- 1 To build roads around the imported assets, toggle the display of aerial imagery assets. On the

View menu, select **Aerial Imagery (F4)**. Then, select the **Road Plan Tool**  and draw a road over the existing road imagery.



- 2 To match the elevation of the road to the imported elevation asset, in the toolbar on the left side of the canvas, click the **Project Roads** button .
- 3 Use the **2D Editor** to view and adjust the elevation profile of the road.

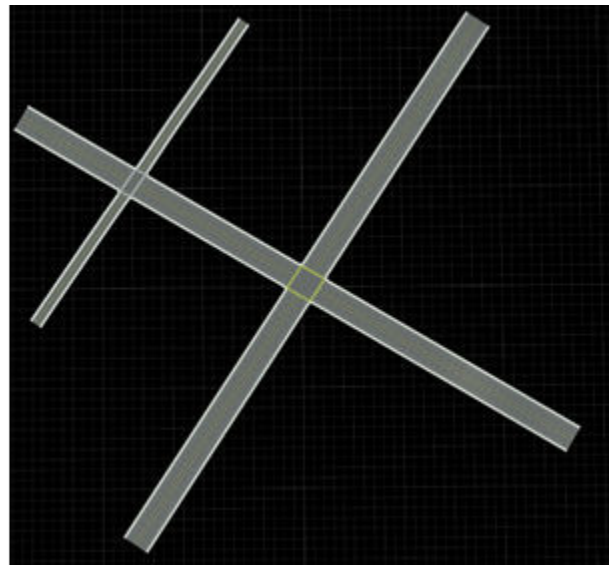
For example, these images show the road elevation (red) being adjusted to match the imported elevation data (blue). You can also adjust the point cloud data (green) to match the elevation data.



To manually control the position or height of the road, in the **2D Editor**, select the appropriate road centers shown in the previous image by purple dots and adjust them to the desired height. To adjust multiple road centers to the same height or position, press **Ctrl+A** and select the points to maneuver them to the appropriate height or position.

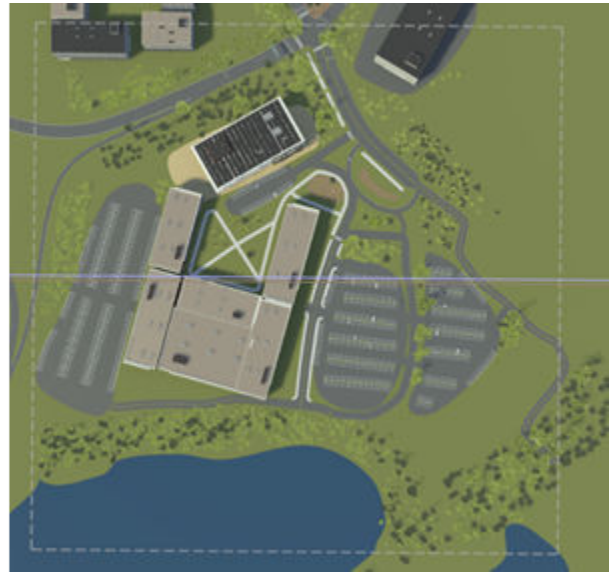
Compare Roads Against Imported GIS Assets

Compare the imported GIS asset data with the road network that you created to check the accuracy of road mapping and synchronization. You can toggle the imported GIS asset from the **View** menu or by using the **Aerial Imagery (F4)**, **Elevation (F5)**, and **Point Cloud (F6)** shortcut keys.



To further customize the scene, you can use different assets to add 3D models, textures, road signs, stencils, and other data shared by multiple RoadRunner scenes. See “RoadRunner Asset Types” on page 2-37.

The left image shows the **Aerial Imagery** view, and the right image shows the corresponding RoadRunner scene. This scene was designed using different tools within RoadRunner to create buildings, roads, parking lots, signboards, trees, and other scene objects.



Create Roads Automatically from HERE HD Live Map Road Data

To automatically generate 3D road models, you can use the “RoadRunner Scene Builder Add-On” product, which imports and automatically synthesizes 3D road models from HERE HD Live Map road data by using the **Scene Builder Tool**. RoadRunner Scene Builder requires a separate license.

See Also

Related Examples

- “Import Data”
- “GIS Data Resources for RoadRunner” on page 3-7

External Websites

- [Here Technologies](#)

RoadRunner Fundamentals

- “RoadRunner Project and Scene System” on page 2-2
- “Window Layouts” on page 2-6
- “Coordinate Space and Georeferencing” on page 2-10
- “Manipulate Scene Objects” on page 2-14
- “Keyboard Shortcuts and Mouse Actions for RoadRunner” on page 2-23
- “Choose a RoadRunner Tool” on page 2-27
- “RoadRunner Asset Types” on page 2-37
- “Create, Import, and Modify Assets” on page 2-42
- “Resolve Geometry Issues” on page 2-48
- “Point Editing” on page 2-52
- “Curve Editing” on page 2-53
- “Polygon Editing” on page 2-55
- “Tangent Editing” on page 2-57
- “Span Editing” on page 2-62
- “Region Graph Editing” on page 2-65
- “Graphics and Startup Issues” on page 2-68
- “Obtain RoadRunner Log Files” on page 2-72

RoadRunner Project and Scene System

In RoadRunner, a project contains assets that are shared by multiple RoadRunner scenes. You can create many scenes within the same project, and the scenes can share assets within the project.

Project assets include various components that are created from files such as 3D models, texture maps, and vector graphics. They also include various files specific to RoadRunner, such as materials and road marking styles.

Projects

When you run RoadRunner, you must select or create a project. The current project is displayed on the title bar. The current project is always the active project in RoadRunner.

Create New Project

- 1 Open the RoadRunner application, and on the **Scene** tab of the start page, click **New Scene**.
- 2 In the Select a Project window, click **New Project**, and browse for the folder in which you want to create the project. To create a project in a folder, the folder must be empty.

Note If you create a project that is on a network drive, changes made to asset files might not be automatically reflected in the **Library Browser**. In addition, performance might be slower. For improved performance and full **Library Browser** functionality, create a project on a local disk.

- 3 Select the assets you want to install with the project. By default, RoadRunner projects include a small assortment of materials, models, and other assets. You can also separately purchase the “RoadRunner Asset Library Add-On”, which comes with a large array of generic and country-specific assets.
 - If you have a RoadRunner Asset Library license, click **Yes** to populate the **Assets** folder with the library assets.
 - If you do not have a license, this option is disabled. Instead, click **No** to populate the **Assets** folder with a set of default assets included with RoadRunner.

RoadRunner opens to a new scene that is created within the new project. The project contents are accessible from the folder in which you created the project.

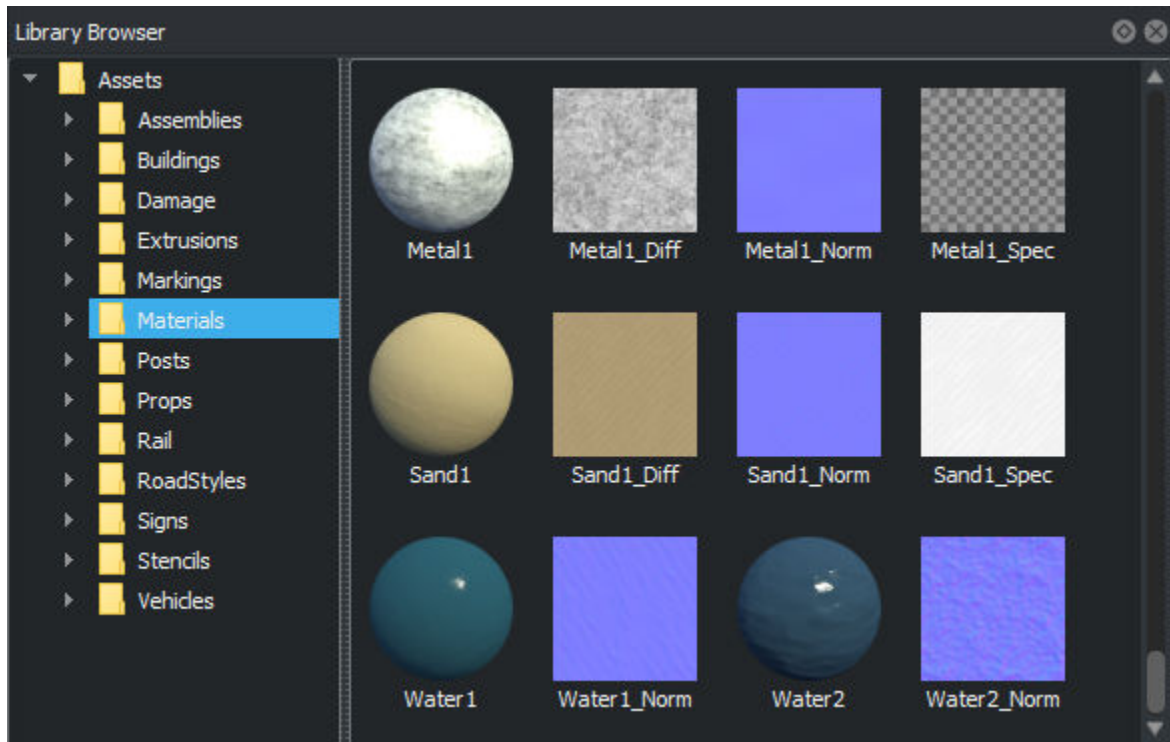
Note It is recommended that you place the entire project folder under version control. For details, see the “Project and Scene Version Control” on page 2-4 section.

Project Folder Contents

A project folder contains these subfolders:

- **Project** — This folder contains a single file named `Project.rrproj`, which defines a unique reference ID for the project. Do not modify or move this file.
- **Scenes** - This folder stores individual scenes that use this project. When you first create a project, this folder contains several sample scenes. When you save a new scene for the current project, RoadRunner defaults to this folder.

- **Exports** - This folder is the default location to write out exported data from RoadRunner. When you first create a project, this folder is empty. You are not required to use this folder for exported data. For more information on exporting data, see “Export Scenes”.
- **Assets** - This folder stores all asset files available for use in a scene. This folder and its subfolders appear in the **Library Browser**.



For every asset within the **Assets** tree, RoadRunner automatically creates an associated metafile with the `.rrmeta` file extension. The metafile contains additional data associated with the asset, the details of which vary for different asset types. The metafile also contains a unique ID, which you can use to identify a specific asset. Even if you rename an asset or move it into a different folder within the **Assets** folder tree, this ID does not change.

Always keep this metafile in the same location as the asset itself. Asset operations performed with the **Library Browser** automatically update the corresponding metafiles. For more details on working with assets, see “Create, Import, and Modify Assets” on page 2-42.

Change Current Project

To switch to a new project when you already have one open in RoadRunner, from the menu bar, select **File** and then **Change Project**. Then, in the Select a Project window, create a new project, browse for the root folder of a different project, or select a recently opened project from the **Recent Projects** list.

Save Project

To save a project, from the menu bar, select **File** and then **Save Project**. In addition, saving a scene also saves any changes you made to the current project. Modified assets are not saved until the current project is saved.

Scenes

A scene file contains an area that includes objects such as roads, surfaces, props, and other scene aspects. It is the main type of file edited in RoadRunner. Scenes can represent anything from a small area, such as a single intersection, to a large area, such as a portion of a city. A scene can contain multiple roads, intersections, road markings, props, terrain sections, and so on.

Individual scenes are saved as `.rrscene` files, typically in the **Scenes** folder of a project. You can create many scenes within the same project, and the scenes can share assets within the project.

RoadRunner has exactly one scene active at any given time. The name of the current scene is displayed in the title bar. If the scene has not yet been saved, the title bar displays the scene name as **New Scene**.

If you have unsaved changes in your current scene, RoadRunner prompts you to save your current scene before starting a new scene, loading an existing scene, or exiting the program.

Create New Scene

To create a new scene while working in an existing scene, on the menu bar, select **File** and then **New Scene**. Alternatively, press **Ctrl+N**.

To create a new scene from the start page, follow these steps:

- 1 On the **Scene** tab of the start page, click **New Scene**.
- 2 Select a previous project or click **New Project** to create your scene in a new project.

Open Existing Scene

To open an existing scene that was recently opened, open it directly from the **Recent Scenes** menu under **File**. Alternatively, select **File** and then **Open**, or press **Ctrl+O**, and then browse for the scene you want to open.

If the selected scene file is in the **Scenes** subfolder of a project folder, that project is loaded automatically. If your scene is saved elsewhere, the scene stores the relative path to the project directory.

If RoadRunner is unable to find or load the project, the software provides the option of selecting a previous project or browsing for a different one.

Save Scene

To save a scene, from the menu bar, select **File** and then **Save Scene**. Saving a scene also saves any changes you have made to the current project. Modified assets are not saved until the current project is saved.

Project and Scene Version Control

When multiple people are making changes to project assets or scenes, using a revision control system such as Git™ to control file versions. With a revision control system, follow these policies:

- Manage files, including metafiles (`.rrmeta`), within the project tree with the revision control system.

- Do not merge any RoadRunner internal files because they are binary files.
- Make sure only one user makes changes to any particular file at any time.

See Also

Related Examples

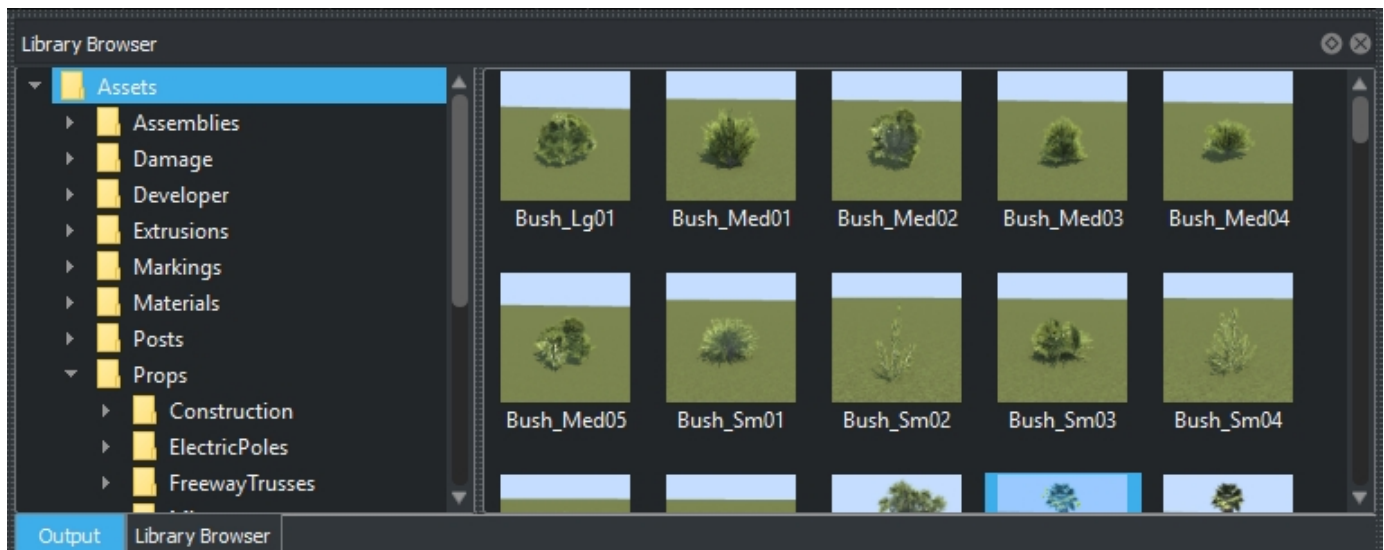
- “Create, Import, and Modify Assets” on page 2-42
- “Create Simple RoadRunner Scene” on page 1-19
- “RoadRunner Asset Types” on page 2-37

Window Layouts

The RoadRunner user interface is organized into panes. You can customize the layout of the pane to meet your preferences by resizing the panes and moving them into different configurations. RoadRunner preserves the window layout from session to session.

You can save and name up to five different layouts. You can restore saved layouts by using the layout controls at the bottom of the window menu.

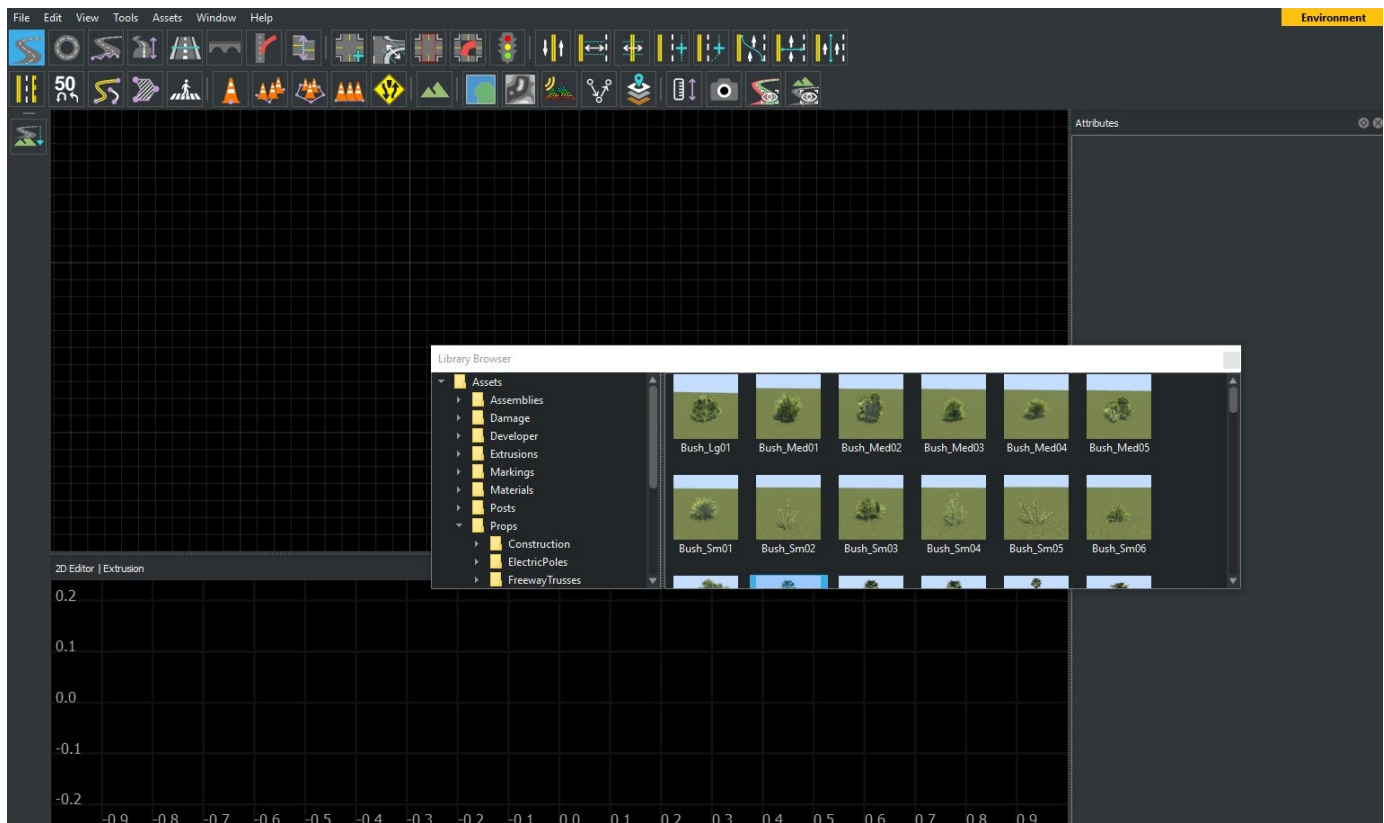
Switch Between Tabbed Panes



Panes can be stacked on top of each other. By default, the **Output** pane and **Library Browser** pane are tabbed.

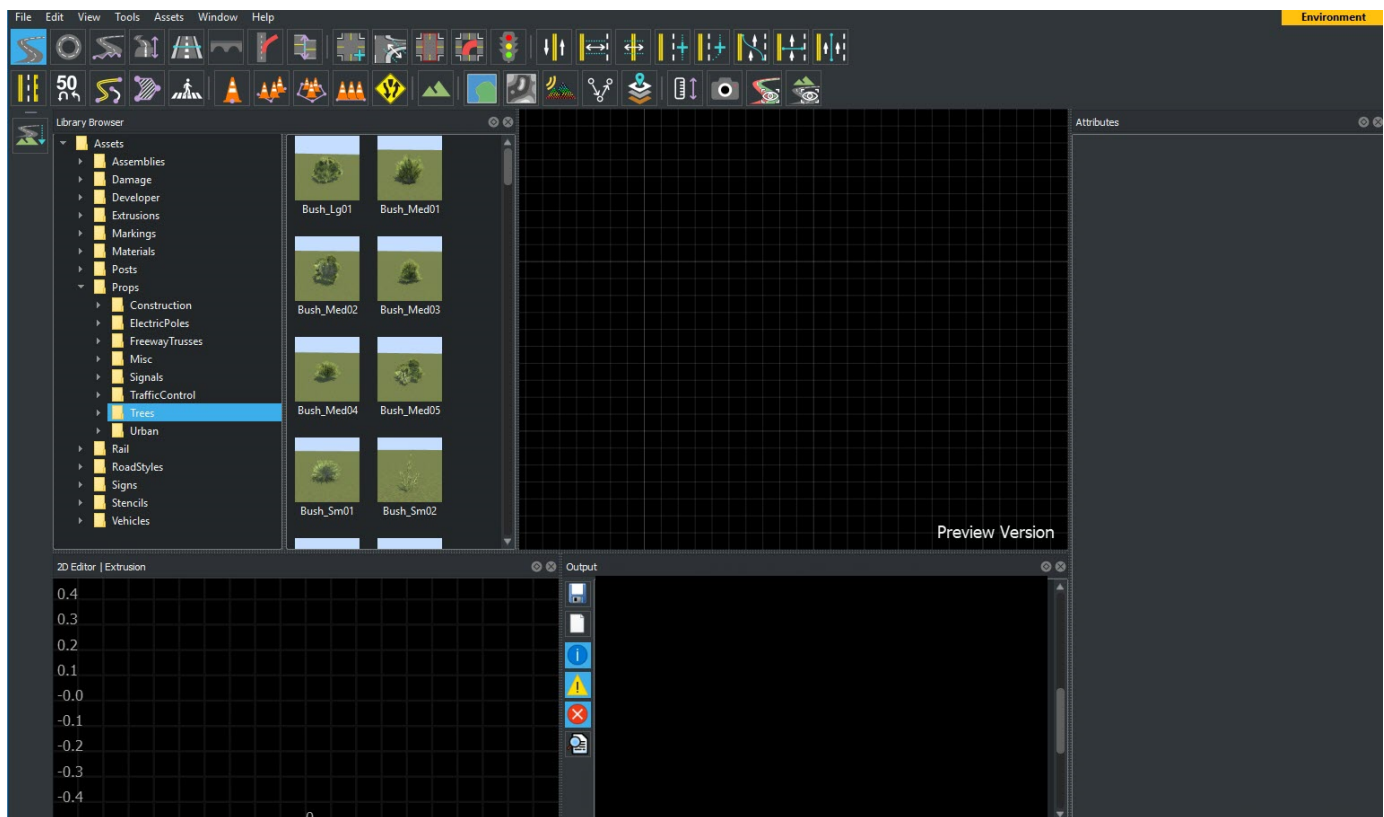
To switch between tabbed panes, click the applicable tab at the bottom of the pane.

Undock a Pane



You can move a pane to a separate window (for example, to move it to a different monitor) by clicking and dragging the top of the pane. After moving a pane to a new window, you can move and resize this pane independently from the main application.

Dock a Pane



You can change where panes are docked in the application. This option can be useful for making better use of screen real estate on wide monitors.

- 1 Click and drag the top of the pane (this action works for docked and undocked panes).
- 2 Hover over the edge of the application where you would like to dock the pane.
 - If you hover over the edge of the application without an existing pane, the pane will be docked to that edge.
 - If you hover over the middle of another pane, the pane will be docked on top of that pane (that is, the panes will be tabbed).
 - If you hover over the left or right side of a bottom pane (or the top or bottom of a side pane), then both pane will be displayed next to each other.

Save the Current Window Layout

- 1 Select the **Window > Save Layout** menu option. A dialog box prompts you to name the layout. If you already have five layouts saved, saving another layout will replace the oldest saved layout.
- 2 Type in the desired name of the layout. If you type in the name of an existing layout, the new layout will replace that existing layout.

Restore a Saved Window Layout

Select the **Window > Apply Layout > (layout)** menu option, where (layout) is the name of the layout you want to restore.

Delete a Saved Window Layout

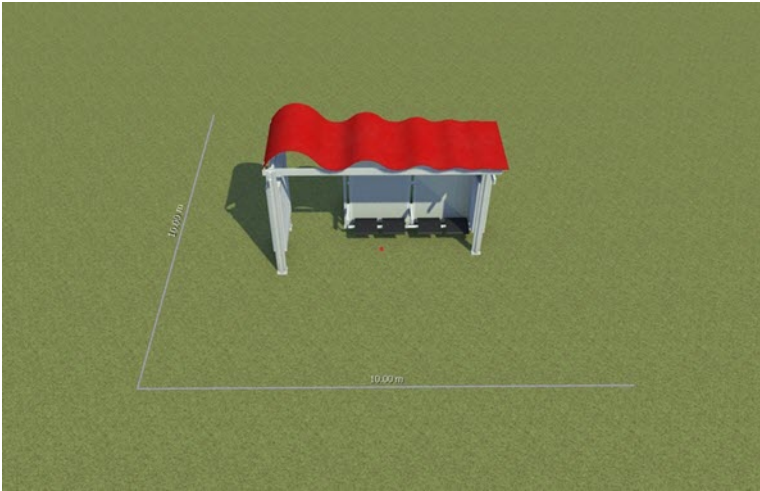
Select the **Window > Delete Layout > (layout)** menu option, where (layout) is the name of the layout you want to delete.

Reset the Window Layout to the Default Layout

Select the **Window > Reset Layout** menu option.

Coordinate Space and Georeferencing

Local Coordinate System



3D coordinates are displayed and edited in a right-handed Cartesian coordinate system. All spatial units are represented in meters, and angles are represented in degrees.

The 'X' and 'Y' dimensions represent 'Easting' and 'Northing' directions, respectively. The 'Z' dimension is height.

This table illustrates local object transformations along each axis (for example, when using the **Prop Point Tool**). Each image shows a transformation of the prop in the image in the positive direction for each axis.

| | X (Easting) | Y (Northing) | Z (Height) |
|--------|-------------|--------------|------------|
| Move | | | |
| Rotate | | | |
| Scale | | | |

Georeferencing (Geographic Coordinates and Projections)



RoadRunner scenes can be optionally georeferenced, which means that coordinates in the scene can be mapped to locations on the Earth. This mapping is important when you want to model a real-world location by using GIS reference data. For more details, see “Import Data”.

Georeferencing Basics

Georeferencing is a varied and complex topic. RoadRunner hides most of this complexity, especially if you are using well-formed GIS reference data.

In many cases, georeferencing data is carried through when exporting. If you want to align exported data with other GIS data (such as a GPS trace), then a basic familiarity with geospatial transformations is required.

To perform geospatial coordinate transformations, RoadRunner uses the PROJ library, which is a robust and industry-standard library for transforming horizontal and vertical coordinate systems. If you need to work with georeferenced data in your own application stack, you can use PROJ for optimal robustness and compatibility (or use a library that uses PROJ internally, such as GDAL or PDAL).

Georeference a Scene

To add or modify a scene's location on the Earth, use the **World Settings Tool**. An initial location is also applied automatically when first dragging any GIS asset into a nongeoreferenced scene.

Georeferenced Coordinate System

RoadRunner supports a variety of input projections and datums when loading external GIS data. However, all editing and displaying is performed in a specific georeferenced coordinate system.

Any external GIS data is transformed automatically into this coordinate system before it is displayed.

Horizontal Georeferenced Coordinate System

To map the X and Y coordinates of the Local Coordinate System on to the Earth, an application must define a horizontal coordinate system (typically by defining a geospatial projection and datum).

RoadRunner uses a coordinate system that reduces scale and rotational distortion surrounding (within ~100 km of) a latitude/longitude point of interest. You can control the latitude/longitude point (using the **World Settings Tool**), but control over the projection is not permitted.

Specifically, RoadRunner uses a Transverse Mercator projection (with a scale factor of 1.0) over the WGS84 datum. For example, a scene centered at a latitude of 32.0 and a longitude of -118 has a horizontal georeferenced coordinate system defined as (in Proj syntax):

```
+proj=tmerc +lat_0=32.0 +lon_0=-118.0 +k=1 +x_0=0 +y_0=0 +datum=WGS84 +units=m
```

Vertical Georeferenced Coordinate System

To map Z (height) coordinates of the Local Coordinate System on to the Earth, an application must define a vertical coordinate system.

Roadrunner uses heights over the EGM96 Geoid, as defined by a 15-minute grid (such as the one found here). Grid files are used to convert between WGS84 ellipsoid heights and geoidal heights.

You can find the exact grid file used by RoadRunner by searching for the "egm96_15.gtx" file in the RoadRunner installation directory.

The vertical coordinate system is also defined in the PROJ string. The full PROJ string for the example in the horizontal section above is:

```
+proj=tmerc +lat_0=32 +lon_0=-118 +k=1 +x_0=0 +y_0=0 +datum=WGS84 +units=m +geoidgrids=egm96_15.gtx +vunits=m +no_defs
```

Georeferencing and Exported Data

In addition to the information in the Georeferencing Basics section, this section provides information about georeferencing information in data exported from RoadRunner.

To align data exported from RoadRunner with other GIS data (or to transform between latitude/longitude coordinates and coordinates in the export data), you must know the projection and datum of each data source. RoadRunner expresses this information as a Proj syntax or WKT string.

Many export formats also include projection information. For example, OpenDRIVE on page 5-15 data exported from RoadRunner includes a <georeference> tag defining the projection information as a PROJ string.

Note In almost all cases, it is not possible to align two georeferenced data sets by simply shifting them. Projection transformations are more complicated than a simple shift and scale. Instead, rely on a library like PROJ.

RoadRunner exports data in the same georeferenced coordinate system used by the scene (see Georeferencing Basics). You can view the PROJ/WKT strings for the current scene in the **World Settings Tool**. Control for transforming data into a different projection during export is not supported.

Exported Data and Grid Files

The exported data also uses the same vertical coordinate system as the scene itself (see Georeferencing Basics). To interpret the elevations in the exported data, you might need to make use of the same grid files used by RoadRunner. This might require supplying the grid files to your external application (if not already present).

In some specific cases, you might be able to ignore the grid files. Examples of these types of cases include if you do not need to vertically match exported RoadRunner data and external GIS data or if all of your external GIS data is already using the same vertical datum as the RoadRunner scene.

If you are confident that you do not need the grid files in your external application, you can remove the `+geoidgrids=egm96_15.gtx` portion of the PROJ string in any exported data. Because there

can be upwards of a 30 m vertical difference between geoidal heights and ellipsoidal heights, if you are confident in how the data is to be used downstream.

Manipulate Scene Objects

In RoadRunner, the tool you select can affect which objects in a scene that you can select, move, create, delete, or modify.

Select Objects

You can select objects in the scene editing canvas and in the **2D Editor** pane.

Most operations in RoadRunner require selecting one or more objects to act on. The attributes of the selected objects are displayed in the **Attributes** pane. Many operations, such as those in the left toolbar for a selected tool, apply to the currently selected objects.

The current tool defines which types of objects are selectable. For example, the **Road Plan Tool** permits the selection of roads but not props, whereas the **Prop Point Tool** permits the selection of prop points but not roads.

Some types of objects can be selected only after first selecting a parent object. For example, in the **Road Plan Tool**, you must first select a road before the control points for that road are displayed.

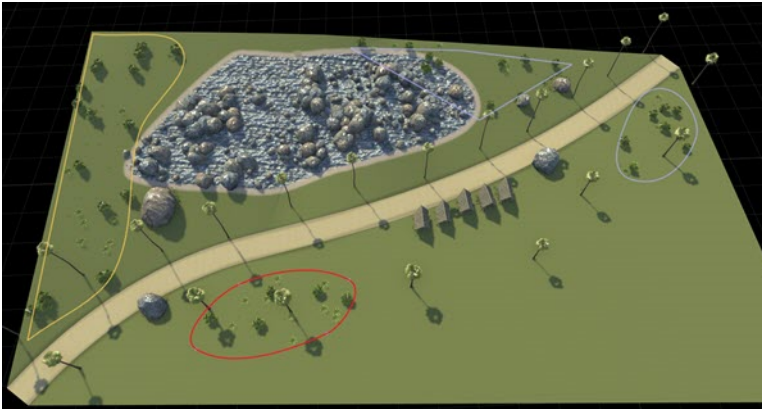
RoadRunner enables you to select multiple objects together. Some tools permit the selection of multiple different types of objects at once.

This screenshot of a simple scene was taken in the **Prop Polygon Tool**. The scene contains four prop polygons, which are displayed as light purple outlines. The sections that follow use images from this scene to show how object selection works.



Selection Colors

Most tools in RoadRunner use a common color language to indicate the selection state. This image shows polygons in three different selection states.



- The light purple polygons are objects that are not currently selected.
- The red polygon contains the objects that are currently selected.
- The yellow polygons contain the objects that the mouse is pointing to. This state provides a visual indication of the object that will be selected if you click.

A fourth color, gray, is also used when selecting overlapping objects. See [Cycle-Select Overlapping Objects](#).

Select Single Object

To select an object, click the object in the scene. This action deselects any previously selected objects and selects the object the mouse is pointing to. In this image, the selection that was previously being pointed to has been clicked and is now selected.



Add Object to Selected Objects

To select an additional object, hold **Shift** and click an unselected object in the scene. In this image, an additional object has been added to the previous selection.



To remove an object from selection, hold **Ctrl+Shift**, and then click a selected object in the scene. This action removes that object from the selected objects, leaving the remaining objects selected.

Alternatively, to add or remove selected objects, hold **Shift** or **Ctrl+Shift** and perform a box select.

Box Select a Group of Objects

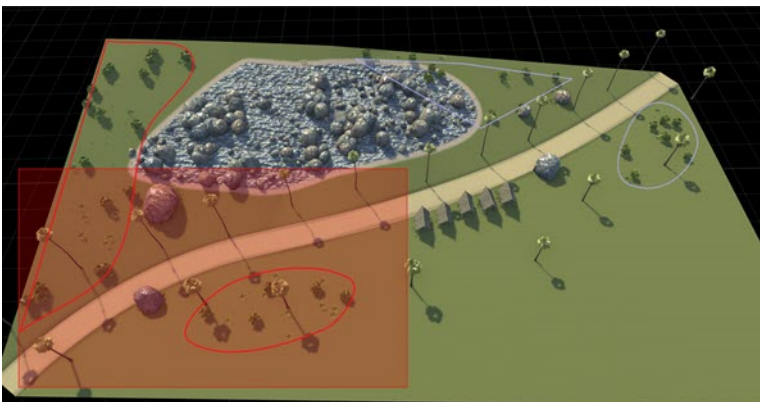
To perform a box selection, in the scene editing canvas, click and drag to draw a rectangle around the objects you want to select. There are two box selection options:

- Overlap Box Selection — Select any objects that touch the box.
- Containment Box Selection — Select only objects that are fully contained within the box.

The direction in which you draw the box dictates which selection type is used. This table indicates the selection type according to the direction in which you draw the box.

| | |
|---------------------------|----------------------------|
| Toward top-left | Toward top-right |
| Containment | Overlap |
| Toward bottom-left | Toward bottom-right |
| Overlap | Overlap |

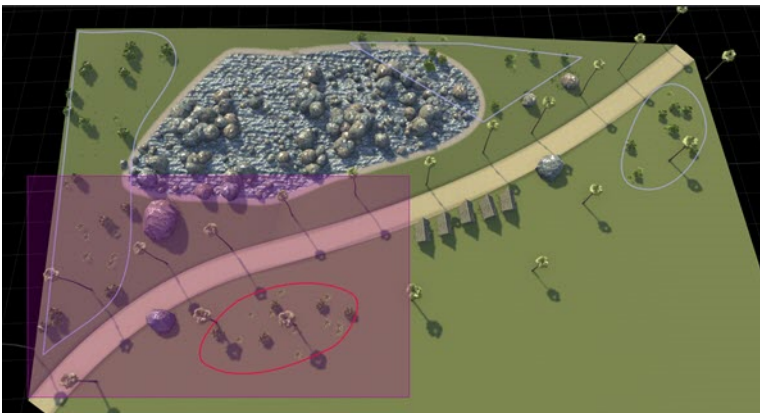
In this image of overlap box selection, both polygons that are at least partially within the region are selected.



To perform an overlap box selection, click and drag in one of the directions indicated by the red boxes in the previous table. A red box appears, and any objects touching that box are selected. Optionally, hold **Shift** to add the objects to the selected set. Hold **Ctrl+Shift** to remove the objects from the selected set.

Tip If no box appears, check that you did not click on or inside a draggable object in the scene. If you hold **Shift** prior to the click, a box selection occurs, even if your drag starts outside of a selectable object.

In this image of containment box selection, only the polygon that is fully contained within the region is selected.

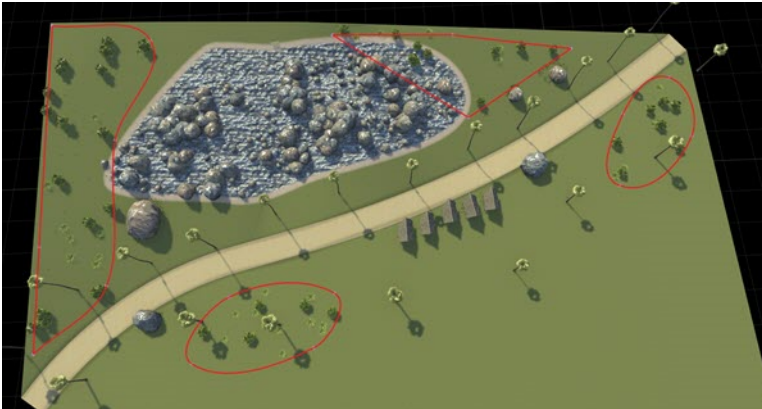


To perform a containment box selection, click and drag in the upper-left direction. A purple box appears, and only those objects fully contained within that box are selected. Optionally, hold **Shift** to add the objects to the selected set. Hold **Ctrl+Shift** to remove the objects from the selected set.

Tip If no box appears, then it is likely that you started the box on or inside a selectable object in the scene. Check that your drag starts outside of a selectable object.

Select All Objects

To select all objects in the scene, from the menu bar, select **Edit** and then **Select All**, or press **Ctrl+A**. This image shows all of the prop polygons selected.



The behavior of a select all action depends on which objects you currently have selected.

- If you have no objects selected, then all selectable objects in the scene are selected.
- If you have objects selected, and any of those objects have unselected child objects, then the unselected child objects are selected. For example, if you select a single prop polygon using the **Prop Polygon Tool**, then the points on that polygon are displayed but are not selected. Performing a select all operation selects all the points on that polygon, not other polygons in the scene.
- If you have objects selected, and all child objects are already selected (or no child objects exist), then all selectable objects in the scene are selected.

Deselect All Objects

To deselect all objects in the scene, from the menu bar, select **Edit** and then **Deselect All**, or press **Ctrl+D**. This image shows all previously selected polygons now deselected.



Cycle-Select Overlapping Objects

Sometimes multiple selectable objects overlap each other. For example, this image shows three overlapping prop polygon objects.



In these cases, you can cycle between the different objects by repeatedly clicking on the overlapping portion. Each click selects the next overlapping object.

- 1 Move the pointer over the area where the object overlap, which in this case is in the middle of the three overlapping polygons. The first object you can select displays in yellow, while the other overlapping objects display in gray.



- 2 Click the overlapping portion to select the object. The selected object displays in red, and the object you can select on the next click displays in yellow.



- 3 Continue clicking to cycle through the overlapping objects until you reach the object you want to select. This image shows the next cycle in the selection.



Move Objects

For tools that enable you to move objects in the scene by selecting and dragging them, follow these steps:

- 1 Select one or more objects.
- 2 Click and drag a selected object to move it. If multiple objects are selected, dragging one object moves all of the selected objects.

The exact behavior when moving objects depends on the specific tool and type of object. For example:

- Props moved in one of the prop tools have their heights automatically projected to the ground surface. For more details on prop tools, see “Props and Signs”.
- Lane marking nodes moved in the **Lane Marking Tool** are constrained to lie along the lane boundary curve.
- Moving a road control point in the **Road Plan Tool** might automatically update other roads to enforce tangential continuity.

Many objects can also be moved during initial creation.

Create Objects

Many tools provide the ability to create objects. The type of object created and the specific creation steps depend on the tool. For step-by-step creation instructions, see the documentation for the specific tool.

In most cases, you can right-click either an existing object or an empty location in the scene to create an object. Often, you can keep holding down the right-click button to simultaneously create and drag the object.

Depending on the specific tool, you might need to first select an appropriate asset in the **Library Browser**. Some tools require an asset to be selected, while others will change their behavior depending on whether or not an asset is selected.

Some types of objects can be created by clicking an asset in the **Library Browser** and dragging the object into the scene. For example, dragging **Prop Model Assets** into the scene adds prop points and automatically switches the currently selected tool to the **Prop Point Tool**.

Delete Objects

Most tools enable you to delete selected objects. You can refer to the documentation for a specific tool to learn about deleting objects by using it, however, the steps are often similar to these ones:

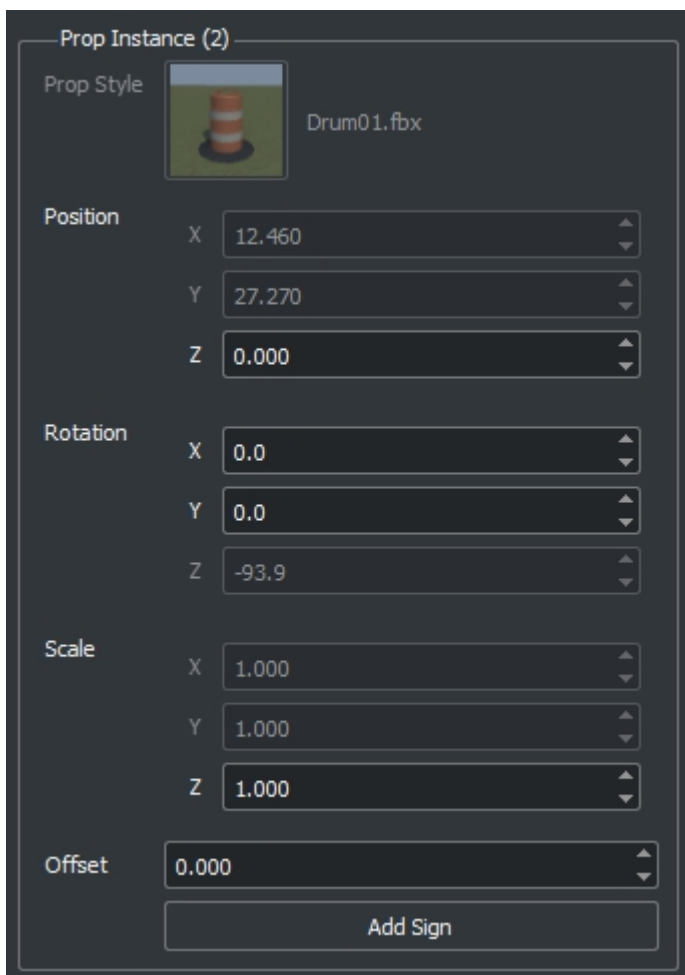
- 1 Select one or more objects.
- 2 Select the **Edit** option and then the **Delete** option in the menu bar, or press **Delete**.

If an object cannot be deleted, the bottom status bar typically displays an error message.

Modify Objects

When you select an object in a scene, you can then modify its properties in the **Attributes** pane. If you select multiple objects, then the **Attributes** pane provides additional features for interacting with them.

This **Attributes** pane shows the selection of two props created by the **Prop Point Tool**.



In this selection:

- The group label, **Prop Instance (2)**, shows the number of selected objects selected, which is two in this case.
- The **Prop Style** asset picker is dimmed, which indicates that the two props have different assets assigned to them.
 - The asset picker shows the style of the first selected object, which in this case is a drum barrel.
 - If the asset picker value changes, then the picker modifies all selected objects and the asset picker is longer dimmed.
- The selected objects have different `Position.X` and `Position.Y` values, but their height (`Position.Z`) value is the same. Modifying the `Position.X` or `Position.Y` values applies the same value to all selected objects.
- Clicking a button, such as the **Add Sign** button in the figure, applies the operation to all selected objects.

To modify an object to use a different asset, select an asset from the **Library Browser** and drag it onto the asset picker image in the **Attributes** pane.

Note Most asset pickers accept only certain types of assets. For example, you can assign **Prop Model Assets** and **Extrusion Assets** to a prop curve by using the **Prop Curve Tool**, but you cannot assign a **Material Assets**.

Some asset pickers enable you to have no asset assigned. To remove an asset from an asset picker, right-click the asset picker and select **Clear**. Alternatively, click the asset picker and press **Delete**.

To quickly locate the currently displayed asset in the **Library Browser**, click the asset picker. The asset is selected in the **Library Browser**.

Although the asset is selected, its attributes do not display. To see and modify the asset attributes, click the selected asset in the **Library Browser**.

See Also

Related Examples

- “Create, Import, and Modify Assets” on page 2-42
- “Choose a RoadRunner Tool” on page 2-27

Keyboard Shortcuts and Mouse Actions for RoadRunner

Scene Editing

These operations apply to the scene editing window and in some cases panes such as the **2D Editor**, **Library Browser**, and **Attributes** panes.

| Task | Action |
|---------------|--|
| Undo or redo. | Ctrl+Z or Ctrl+Y Undo and redo also apply to most selection and tool change actions. RoadRunner has an infinite undo and redo stack. This stack is related to the current scene. Operations that change the current scene, such as opening a different scene or creating a new scene, remove all actions from the stack. |
| Select all. | Ctrl+A |
| Deselect all. | Ctrl+D |
| Delete. | Delete |

Object Selection

For more details about object selection, see “Manipulate Scene Objects” on page 2-14.

| Task | Action |
|------------------------------------|---|
| Select an object. | Click. |
| Add an object to a selection. | Hold Shift , then click. |
| Remove an object from a selection. | Hold Ctrl+Shift , then click. |
| Select multiple objects. | Hold Shift , then right-click and drag to draw a box around the objects you want to select. <ul style="list-style-type: none"> • Drag the box toward the top-left to select only objects that are fully contained in the box. • Drag the box toward any other direction (bottom-left, top-right, bottom-right) to select any object that is at least partially within the box. |

Camera Control (Scene Editing Canvas)

For more details on controlling the camera in the canvas, see “Camera Control in RoadRunner” on page 1-44.

| Task | Action |
|---|--|
| Rotate the camera around a point of interest. | Hold Alt , then click and drag. Note In Linux, Ubuntu 16.04, pressing the Alt key moves the current window and pressing the Windows key shows certain help overlays. To use the Windows key instead of the Alt key for moving windows, update Ubuntu 16.04 according to the instructions in “Update Linux Ubuntu Key Mapping” on page 2-25. |
| Zoom in or out. | Hold Alt , then right-click and drag up or right to zoom in, or down or left to zoom out. Alternatively, move the mouse scroll wheel up or down. |
| Pan across the scene in the x-direction or y-direction. | Hold Alt , left-click, and right-click, then drag. Alternatively, hold the middle-mouse button, then drag. |
| Frame the camera around an object of interest. | Select an object and press F . |
| Switch to perspective view, where distant objects appear smaller than close objects. | Press P . |
| Switch to orthographic view, where objects do not change size as they get closer or farther away. | Press O . |
| Set the camera view to point top down. | Press 5 on the number pad. |
| Set the camera view to point north. | Press 8 on the number pad. |
| Set the camera view to point south. | Press 2 on the number pad. |
| Set the camera view to point west. | Press 4 on the number pad. |
| Set the camera view to point east. | Press 6 on the number pad. |

Camera Control (2D Editor)

| Task | Action |
|---|--|
| Zoom in or out. | Hold Alt , then right-click and drag up or right to zoom in, or down or left to zoom out. Alternatively, move the mouse scroll wheel up or down. |
| Pan across the scene in the x-direction or y-direction. | Hold Alt , then click and drag. Alternatively, hold the middle-mouse button, then drag. |
| Frame the camera around an object of interest. | Select an object and press F . |

Scene Views

Use these keyboard shortcuts to show or hide various scene aspects on the canvas.

| Task | Action |
|------------------------|------------|
| View shaded wireframe. | F3 |
| View aerial imagery. | F4 |
| View elevation map. | F5 |
| View point cloud. | F6 |
| View vector data. | F7 |
| View scene. | F8 |
| View OpenDRIVE data. | F9 |
| View transfer map. | F10 |

Utilities

| Task | Action |
|---------------------------|---------------|
| Take screenshot of scene. | Ctrl+P |

File Operations

| Task | Action |
|-------------------|---------------------|
| Create new scene. | Ctrl+N |
| Open scene. | Ctrl+O |
| Save scene. | Ctrl+S |
| Save scene as. | Ctrl+Shift+S |
| Exit RoadRunner. | Alt+F4 |

Update Linux Ubuntu Key Mapping

In Linux, Ubuntu 16.04, pressing the **Alt** key moves the current window, and pressing the **Windows** key shows certain help overlays. It is recommended that you update Ubuntu to use the **Windows** key (instead of the **Alt** key) for moving windows. To make this update, follow these steps:

- 1 Install `dconf-editor` if it is not already installed. At the command line, enter this code:


```
sudo apt-get install dconf-editor
```
- 2 Open `dconf-editor`.
- 3 Navigate to **org > gnome > desktop > wm > preferences**.
- 4 Change the `mouse-button-modifier` to `<Super>`.

Note It is important to assign a valid key to `mouse-button-modifier`. Leaving that option blank prevents the mouse from interacting with any windows.

See Also

More About





- “Create Simple RoadRunner Scene” on page 1-19
- “Camera Control in RoadRunner” on page 1-44
- “Manipulate Scene Objects” on page 2-14






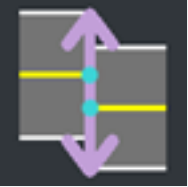
Choose a RoadRunner Tool

The RoadRunner toolbar contains a variety of tools for editing scenes. The tool that you select determines the objects that you can select and edit in the scene editing canvas. In addition, certain actions automatically change the active tool. For example, dragging certain types of assets from the **Library Browser** into the scene adds that asset to the scene and automatically switches RoadRunner to an appropriate tool for that asset.

Road Tools



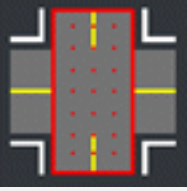


Road tools are used to create and modify roads and their attributes.

| Tool | Description |
|--|--|
| <p>Road Plan Tool</p>  | <p>Create and lay out roads.</p> |
| <p>Road Circle Tool</p>  | <p>Build closed circular loop road, such as for creating roundabouts.</p> |
| <p>Cross Section Tool</p>  | <p>Manipulate banking, crowning, and curb shapes at road cross-sections.</p> |
| <p>Road Height Tool</p>  | <p>Manipulate vertical profile of roads.</p> |

| Tool | Description |
|--|--|
| <p>Road Superelevation Tool</p>  | <p>Adjust superelevation (slope and banking angle) for full width of road.</p> |
| <p>Road Chop Tool</p>  | <p>Chop single road into two connected roads.</p> |
| <p>Road Construction Tool</p>  | <p>Specify physical construction of road sections.</p> |
| <p>Road Speed Limits Tool</p>  | <p>Set speed limits along road sections.</p> |
| <p>Slip Road Tool</p>  | <p>Create onramps, offramps, and road splits.</p> |
| <p>Road Offset Tool</p>  | <p>Adjust connection between two end-to-end roads.</p> |







Junction Tools




Junction tools are used to modify junction (intersection) geometry and lane connectivity. In most cases, junctions are initially created using the **Road Plan Tool** or **Slip Road Tool**. In advanced situations, the **Custom Junction Tool** provides more explicit control over junction creation.

| Tool | Description |
|--|--|
| <p data-bbox="240 485 532 516">Custom Junction Tool</p>  | <p data-bbox="862 485 1365 548">Override RoadRunner automatic junction functionality for advanced cases.</p> |
| <p data-bbox="240 716 402 747">Corner Tool</p>  | <p data-bbox="862 716 1438 747">Adjust shape and materials of junction corners.</p> |
| <p data-bbox="240 978 532 1010">Junction Surface Tool</p>  | <p data-bbox="862 978 1446 1041">Control how road elevations and cross-sections influence interior triangulation of intersections.</p> |
| <p data-bbox="240 1241 443 1272">Maneuver Tool</p>  | <p data-bbox="862 1241 1425 1304">Manipulate individual maneuver roads (paths) within junction.</p> |
| <p data-bbox="240 1503 394 1535">Signal Tool</p>  | <p data-bbox="862 1503 1468 1566">Configure junction signalization and signal traffic phases.</p> |

Lane Tools



Lane tools are used to create and edit lanes and their properties.





| Tool | Description |
|--|--|
| <p>Lane Tool</p>  | <p>Delete lanes and update lane type and travel direction.</p> |
| <p>Lane Width Tool</p>  | <p>Adjust lane widths along road.</p> |
| <p>Lane Offset Tool</p>  | <p>Adjust location of center lane of road.</p> |
| <p>Sidewalk Height Tool</p>  | <p>Modify sidewalk and curb heights.</p> |
| <p>Lane Add Tool</p>  | <p>Add fully formed lanes along road.</p> |
| <p>Lane Form Tool</p>  | <p>Add forming or ending lane along road.</p> |

| Tool | Description |
|---|---|
| Lane Curve Tool  | Create tapering cut in lanes, such as for creating dedicated turn lane in median. |
| Lane Chop Tool  | Cut single lane into two lanes. |
| Lane Split Tool  | Split lane lengthwise into two lanes. |

Marking Tools

Marking tools are used to create and modify road paint, lane markings, and decals in the scene.



| Tool | Description |
|--|--|
| Lane Marking Tool  | Add linear markings to lane boundaries. |
| Marking Point Tool  | Place point markings (stencils), such as arrows and words, on road surfaces. |

| Tool | Description |
|--|--|
| <p data-bbox="240 296 509 327">Marking Curve Tool</p>  | <p data-bbox="862 296 1430 359">Place straight or curved markings at arbitrary locations.</p> |
| <p data-bbox="240 562 537 594">Marking Polygon Tool</p>  | <p data-bbox="862 562 1430 625">Define areas of asphalt patches or repeated marking stripes on roads and terrain surfaces.</p> |
| <p data-bbox="240 829 412 861">Parking Tool</p>  | <p data-bbox="862 829 1458 892">Define parking spaces and other parking-related markings.</p> |
| <p data-bbox="240 1045 643 1077">Crosswalk And Stop Line Tool</p>  | <p data-bbox="862 1045 1474 1108">Add crosswalks and stop lines between corners at intersections.</p> |

Prop Tools


Prop tools are used to create and modify 3D props and road furniture, such as trees, signs, and guardrails.

RoadRunner can import props from a variety of file formats and uses a rich set of tools to place props in the scene. These tools are used to place a variety of different asset types, including **Prop Model Assets**, **Prop Set Assets**, **Sign Assets**, and **Extrusion Assets**.

| Tool | Description |
|--|---|
| <p>Prop Point Tool</p>  | <p>Place individual props in a scene and connect them to other props.</p> |
| <p>Prop Curve Tool</p>  | <p>Place props and extrusions along free-form curves.</p> |
| <p>Prop Polygon Tool</p>  | <p>Place props within arbitrarily shaped regions.</p> |
| <p>Prop Span Tool</p>  | <p>Place props and extrusions along road.</p> |
| <p>Sign Tool</p>  | <p>Modify custom signs, such as street name signs and freeway billboards.</p> |





Terrain Tools




Terrain tools are used to create and modify the surfaces around and between roads. For more details on surfaces, see "How Surfaces Work in RoadRunner" on page 4-4.

| Tool | Description |
|--|--|
| <p data-bbox="240 296 412 327">Surface Tool</p>  | <p data-bbox="862 296 1458 359">Model surfaces around roads, such as walkways, driveways, parking lots, and natural terrain.</p> |

GIS Tools



GIS tools are used to import and adjust a variety of common geographic information system (GIS) files.



| Tool | Description |
|--|--|
| <p data-bbox="240 800 509 831">Aerial Imagery Tool</p>  | <p data-bbox="862 800 1382 863">Manage import and configuration of aerial imagery files.</p> |
| <p data-bbox="240 1020 503 1052">Elevation Map Tool</p>  | <p data-bbox="862 1020 1390 1083">Manage import and configuration of digital elevation model (DEM) files.</p> |
| <p data-bbox="240 1241 467 1272">Point Cloud Tool</p>  | <p data-bbox="862 1241 1442 1304">Manage import and configuration of lidar point cloud files.</p> |
| <p data-bbox="240 1503 467 1535">Vector Data Tool</p>  | <p data-bbox="862 1503 1450 1566">Manage import and configuration of vector data files and explore shape attributes.</p> |

| Tool | Description |
|---|---|
| OpenDRIVE Viewer Tool  | Visualize OpenDRIVE data for import. |
| Scene Builder Tool  | Generate 3D road models from HERE HD Live Map data. |
| World Settings Tool  | Configure geographic position and size of environment model for data import and export. |

Utility Tools

These assorted utility tools enable you to measure distances, capture screenshots, and preview export data.

| Tool | Description |
|--|--|
| Measurement Tool  | Measure positions, distances, and angles in scene. |
| Screenshot Tool  | Generate and save image of current camera view. |

| Tool | Description |
|---|---|
| <p data-bbox="240 296 678 327">OpenDRIVE Export Preview Tool</p>  | <p data-bbox="862 296 1404 359">Visualize and validate OpenDRIVE export of scene and load external OpenDRIVE files.</p> |
| <p data-bbox="240 562 597 594">Scene Export Preview Tool</p>  | <p data-bbox="862 562 1356 594">Preview scene geometry to be exported.</p> |

See Also

Related Examples

- “Point Editing” on page 2-52
- “Curve Editing” on page 2-53
- “Polygon Editing” on page 2-55
- “Tangent Editing” on page 2-57
- “Span Editing” on page 2-62
- “Region Graph Editing” on page 2-65

RoadRunner Asset Types

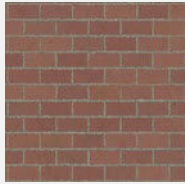

Assets are 3D models, textures, GIS files, and other data that are shared by multiple RoadRunner scenes. You can view and modify assets from the **Library Browser**. For information about how assets are stored in a project, see “RoadRunner Project and Scene System” on page 2-2.

RoadRunner also supports a variety of file formats for developing your own assets. For more details, see “Create, Import, and Modify Assets” on page 2-42.

RoadRunner comes installed with a small library of various assets to get you started. With a “RoadRunner Asset Library Add-On” license, you can install hundreds of additional assets to use in your scenes.

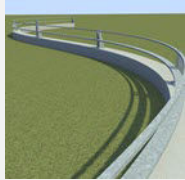
Texture and Material Assets

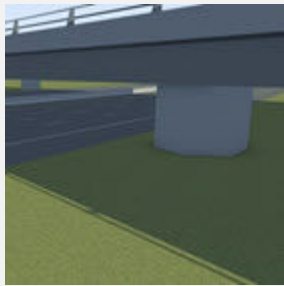




Material assets are used to define the visual properties of surfaces, sidewalks, lanes, and other objects. Texture assets are image files, typically used as texture channels for material assets.


| Asset | Description |
|---|--|
| Texture Assets  | Define texture channels for material assets. |
| Material Assets  | Define visual properties of surfaces, sidewalks, lanes, and other objects. |

Prop Assets

Prop assets define 3D objects that you can place within a scene.




| Asset | Description |
|--|---|
| Extrusion Assets  | Define extruded geometry for features such as walls, guard rails, and fences. |


| Asset | Description |
|---|---|
| <p>Post Assets</p>  | <p>Define building support posts, such as for bridges and overpasses.</p> |
| <p>Prop Model Assets</p>  | <p>Define external 3D model files to add to scene.</p> |
| <p>Prop Assembly Assets</p>  | <p>Define collection of prop instances stored as single asset.</p> |
| <p>Prop Set Assets</p>  | <p>Define collections of props that have a random distribution.</p> |
| <p>Sign Assets</p>  | <p>Define standard and custom street signs.</p> |

| Asset | Description |
|---|--|
| <p data-bbox="240 296 423 327">Signal Assets</p>  | <p data-bbox="862 296 1438 327">Define dynamic traffic signal heads with lights.</p> |

Marking Assets


Marking assets define markings found on roads, such as crosswalks, lanes, and road stencils such as arrows, text, and symbols.

| Asset | Description |
|--|--|
| <p data-bbox="240 890 597 921">Crosswalk Marking Assets</p>  | <p data-bbox="862 890 1458 953">Define crosswalk markings, such as color, width, and spacing.</p> |
| <p data-bbox="240 1148 526 1180">Lane Marking Assets</p>  | <p data-bbox="862 1148 1442 1211">Define lane markings, such as color, width, and dash spacing.</p> |
| <p data-bbox="240 1407 566 1438">Polygon Marking Assets</p>  | <p data-bbox="862 1407 1390 1470">Define space-filling road markings, such as crosshatch and chevron markings.</p> |

| Asset | Description |
|--|--|
| <p data-bbox="240 296 553 327">Stencil Marking Assets</p>  | <p data-bbox="862 296 1453 359">Define road paint features, such as arrows, text, and symbols.</p> |

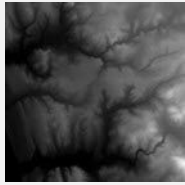
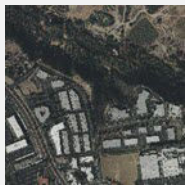
Road Assets



Road style assets are templates that specify the properties of new roads.

| Asset | Description |
|--|--|
| <p data-bbox="240 762 480 793">Road Style Assets</p>  | <p data-bbox="862 762 1445 825">Define templates that specify properties of new roads.</p> |

GIS Assets

Geographic information system (GIS) assets carry georeferencing information, which you can use to position these assets on the Earth. For information on finding GIS data resources that are compatible with RoadRunner, see “GIS Data Resources for RoadRunner” on page 3-7.

| Asset | Description |
|--|---|
| <p data-bbox="240 1291 529 1323">Elevation Map Assets</p>  | <p data-bbox="862 1291 1341 1323">Add GIS raster elevation data to scene.</p> |
| <p data-bbox="240 1545 509 1577">Aerial Image Assets</p>  | <p data-bbox="862 1545 1459 1608">Add GIS satellite and aerial imagery to scene for visual reference.</p> |

| Asset | Description |
|--|---|
| Vector Data Assets  | Add GIS shapefiles and other vector data to scene for visual reference. |
| Point Cloud Assets  | Add aerial or vehicular point clouds to scene for visual reference. |

See Also

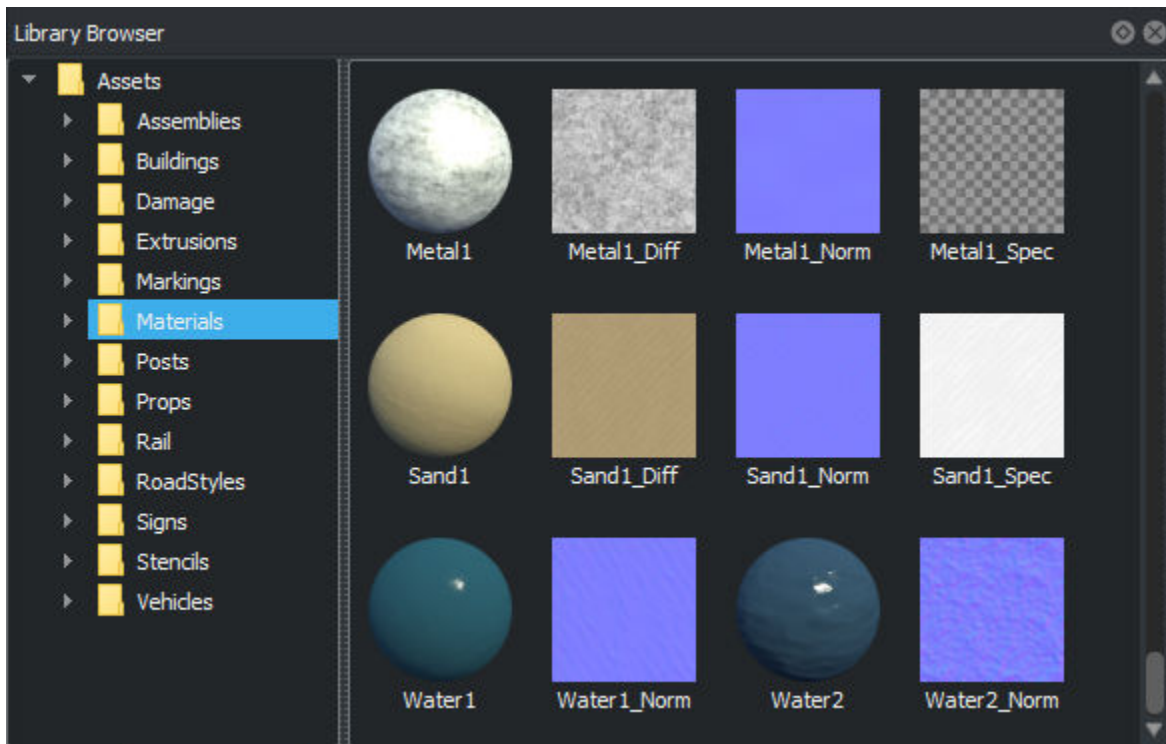
Related Examples

- “Create, Import, and Modify Assets” on page 2-42
- “RoadRunner Project and Scene System” on page 2-2
- “RoadRunner Asset Library Add-On”

Create, Import, and Modify Assets

Assets are the materials, textures, props, and other 3D objects that are available in a RoadRunner project and that can be added to a scene. To browse, create, and modify assets in a current project, use the **Library Browser**. The **Library Browser** is divided into two panes:

- The left pane displays the directory structure within the Assets folder, enabling you to quickly navigate the folder hierarchy.
- The right pane displays the contents of the currently selected folder. When you select an asset from this pane, you can view its attributes from the **Attributes** pane and preview it in the asset viewer.



Using these two panes, you can create and modify assets, manage asset files in your project, and add assets to scenes.

Only files recognized by RoadRunner as assets are displayed. Other system files and auxiliary files in the directory are not shown. For more details on the asset files contained in a project, see “RoadRunner Project and Scene System” on page 2-2.

Create and Import Assets

Depending on the asset type, you can either create new assets for a project either directly in the **Library Browser** or you can import files created outside RoadRunner into the **Library Browser** to create RoadRunner assets.

Create Asset Within RoadRunner

You can create some assets directly within the **Library Browser**, such as materials and road styles. Follow these steps:

- 1 Navigate to the folder in the **Library Browser** where you want to create the new asset.
- 2 Right-click in the **Library Browser** and select **New**, then select **(Asset Type)**. Alternatively, select **Assets**, and then select the **(Asset Type)** menu option.
- 3 Specify a name and press **Enter**.

You can also:

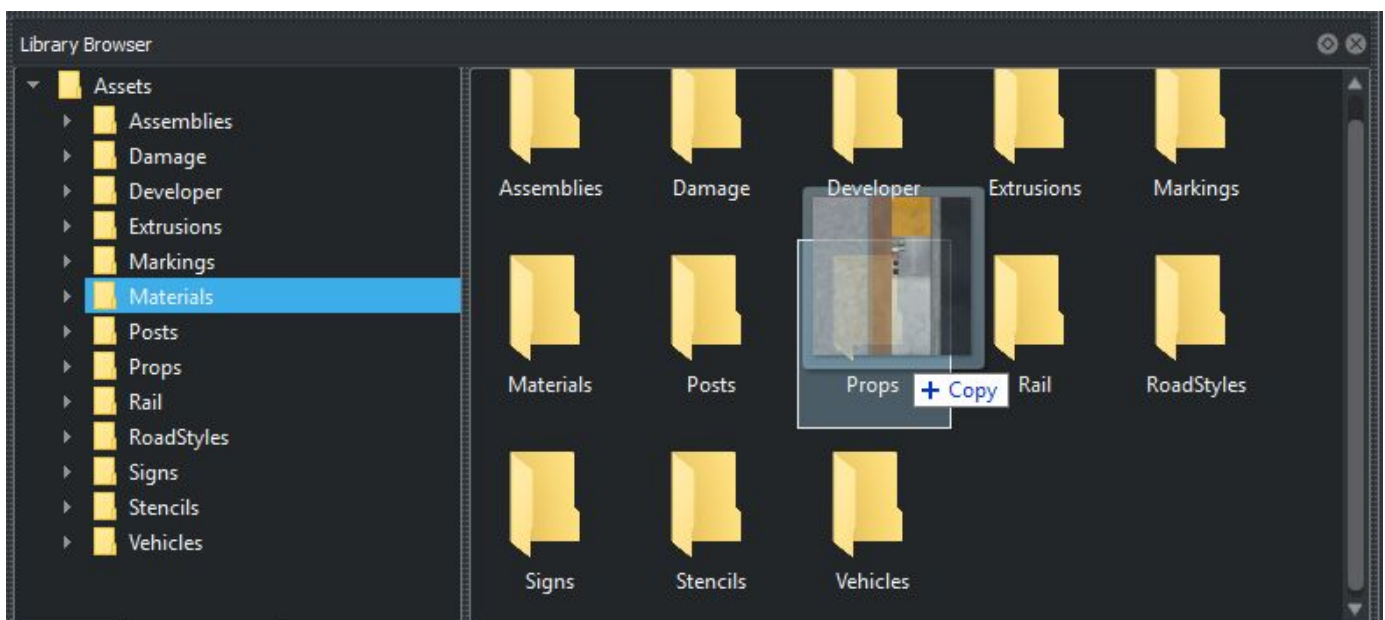
- 1 Right-click the asset in the **Library Browser** and select **Duplicate**.
- 2 Specify a name for the new asset and press **Enter**.

You can use these steps only for assets created within RoadRunner, such as a material or road style asset. If you want to duplicate an asset that depends on another file, such as a texture image or 3D model, duplicate the dependent file only (not the `rmeta` file) by using the file explorer for your operating system.

Create Asset by Importing File Created Outside RoadRunner

Some assets depend on files created outside of RoadRunner, such as a texture image saved as a PNG file or a 3D tree saved as an FBX file. To create such assets for use in RoadRunner, drag the dependent file into the **Library Browser**. Follow these steps:

- 1 Navigate to the folder in the **Library Browser** where you want to add the new asset.
- 2 In the file explorer window for your operating system, navigate to the location of the dependent file (for example, the texture image or the 3D model).
- 3 Select the file (and any associated files or folders) in the file explorer.
- 4 Drag the file, and any associated files or folders, into the **Library Browser**.



This operation copies, rather than moves, the selected files into the directory of the current project.

Alternatively, you can perform these steps by using the file explorer window by moving the files somewhere under the **Assets** directory of your project. This option can be useful if you want to move rather than copy the files, or if you want to use an external script to create assets in a project.

Modify Assets

The steps to modify an asset differ depending on the specific type of asset. For more details, refer to the documentation for the specific asset type.

In most cases, you can modify an asset by following these steps:

- 1 Select the asset in the **Library Browser**.
- 2 View and modify the asset attributes displayed in the **Attributes** pane.

Some assets, such as **Sign Assets**, are modified using the **2D Editor** pane.

Modifications made to an asset are saved only when you next save the project.

Reload Modified Assets

If you change an asset by using an external application, such as modifying a texture file using an image editor, you can force RoadRunner to reload the asset. Right-click in the **Library Browser** and select **Update Assets**. Alternatively, select **Assets** and then the **Update Assets** menu option.

Manage Assets

You can use the **Library Browser** to manage the assets in your project.

Rename Asset

- 1 Right-click the asset in the **Library Browser** and select **Rename** (or press **F2**).
- 2 Specify a new name and press **Enter**.

Move Assets or Folders

- 1 Select assets or folders in the **Library Browser**.
- 2 Click and drag the assets or folders to a different folder in either the left or right pane.

Moving an asset automatically updates asset references in the current scene, but other saved scenes might still reference the old asset location. See “Find Moved Assets” on page 2-45.

Create New Asset Folder

- 1 Right-click in the **Library Browser** and select **New**, then **Folder**. Alternatively, select **Assets**, then **New**, then **Folder**.
- 2 Specify a name and press **Enter**.

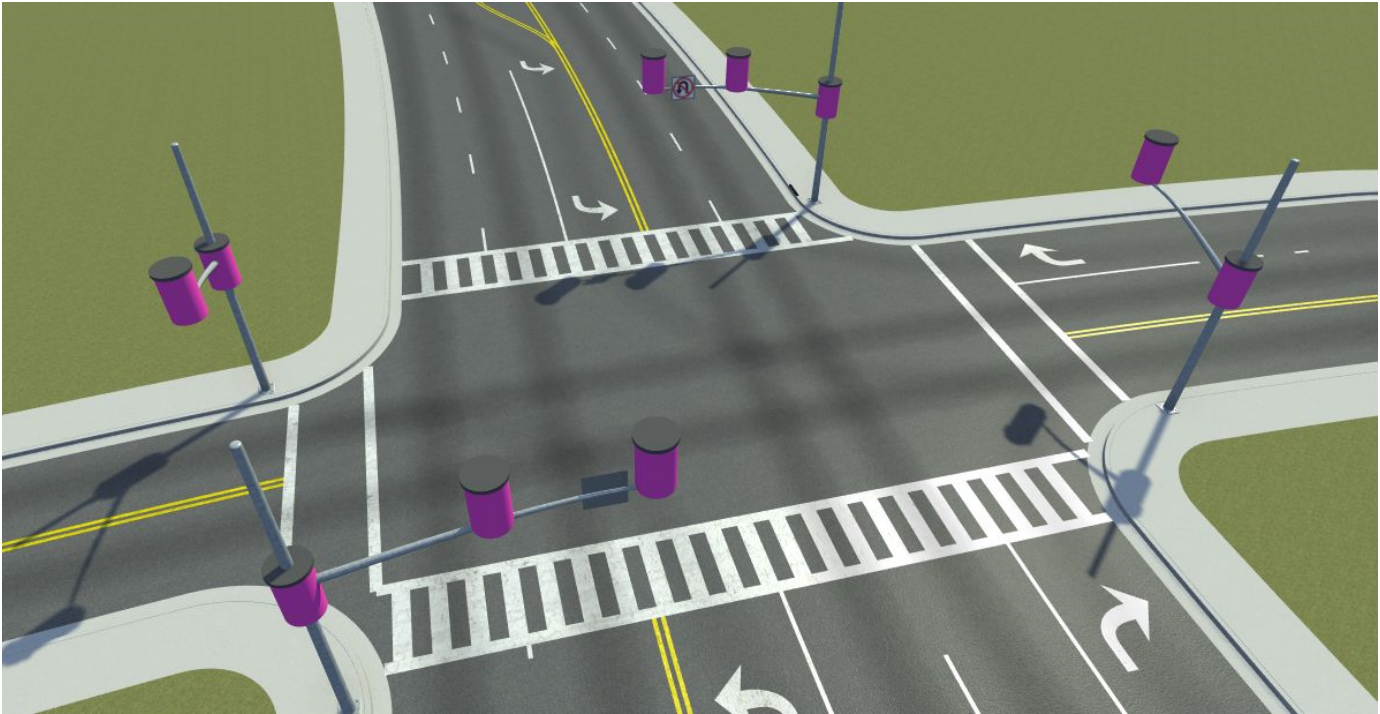
Delete Asset or Asset Folder

- 1 Select assets or folders in the **Library Browser**.

- 2 Right-click and select **Delete**. Alternatively, select the **Edit** , then **Delete** menu option, or press **Delete**.

Find Moved Assets

If an asset or asset folder has been moved or renamed, then existing scene files might still refer to the old location. If a scene cannot find an asset, RoadRunner replaces references to that asset with a visually distinct fallback asset (for example, props display as pink barrels, and textures display as striped red and blue images).



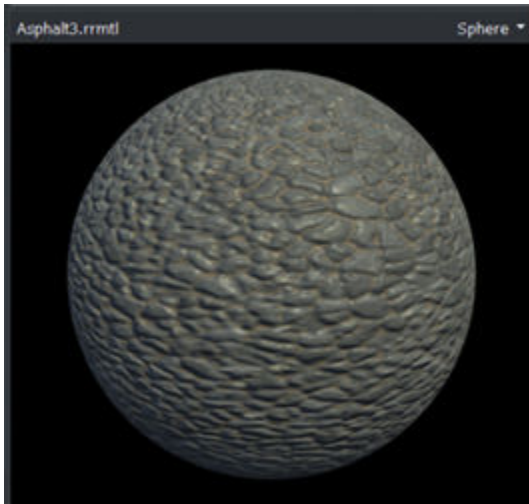
If you encounter this situation, RoadRunner can search for the missing references and attempt to relink them. Follow these steps:

- 1 Open the scene file containing the missing asset references.
- 2 Right-click in the **Library Browser** and select **Update Assets**. Alternatively, select the **Assets** , then **Update Assets** menu option.

This search finds moved assets only if the corresponding `rrmeta` file was also moved or renamed and was left intact.

Visualize Assets

Using the asset viewer, you can visualize the currently selected asset in the **Library Browser**. By default, the asset viewer appears below the **Attributes** pane when a single asset is selected in the **Library Browser**. This image shows a sample material asset.

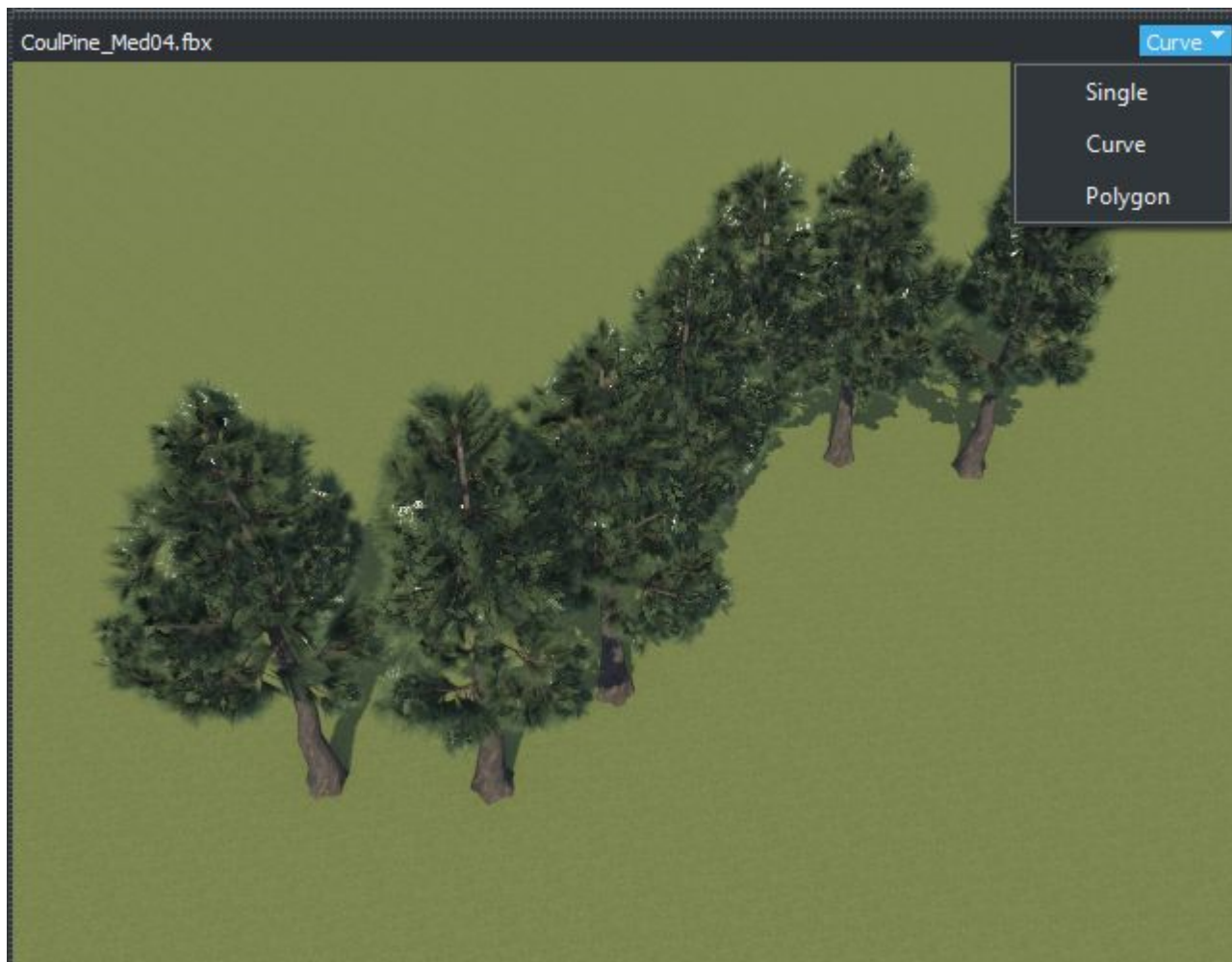


The asset viewer displays different asset types, such as 3D model assets and 2D image assets, in different ways. If the selected asset type supports a 3D display, you can move the camera by using the same controls listed in “Camera Control in RoadRunner” on page 1-44. Unlike the other render windows, you do not need to hold **Alt** to adjust the camera in the asset viewer.

Change Asset Display Type

Some types of assets support additional viewing options. For example, **Material Assets** can be displayed on different types of geometry, and **Prop Model Assets** can be displayed as a point, curve, or collection.

To change the asset display type, in the top-right corner of the asset viewer, click the current display type and select the new display type you want.



Add Asset to Scene

The steps to add an asset to a scene differ depending on the specific tool and type of asset. Refer to “Create Objects” on page 2-20.

Most assets can be added to a scene by dragging them into that scene. Click and drag an asset from the **Library Browser** into the scene editing canvas. RoadRunner automatically switches the current editing tool to the appropriate tool for that asset.

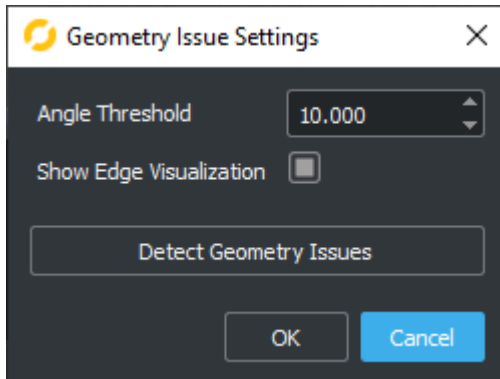
See Also

Related Examples

- “RoadRunner Asset Types” on page 2-37
- “RoadRunner Project and Scene System” on page 2-2
- “Manipulate Scene Objects” on page 2-14

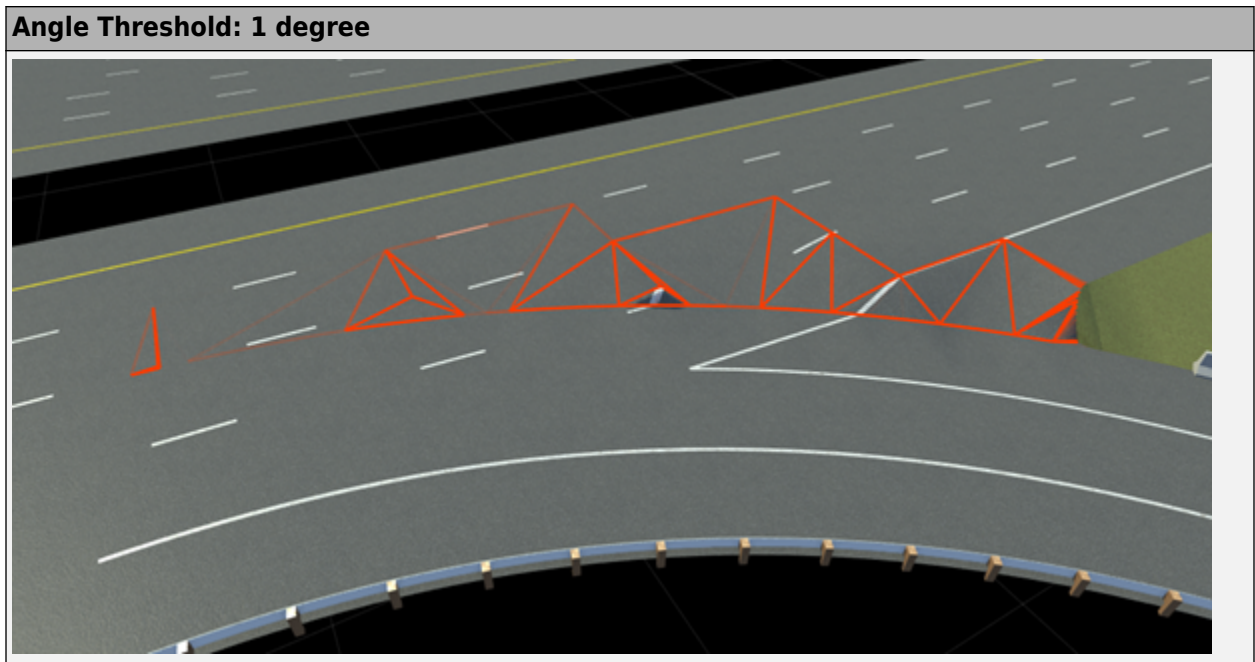
Resolve Geometry Issues

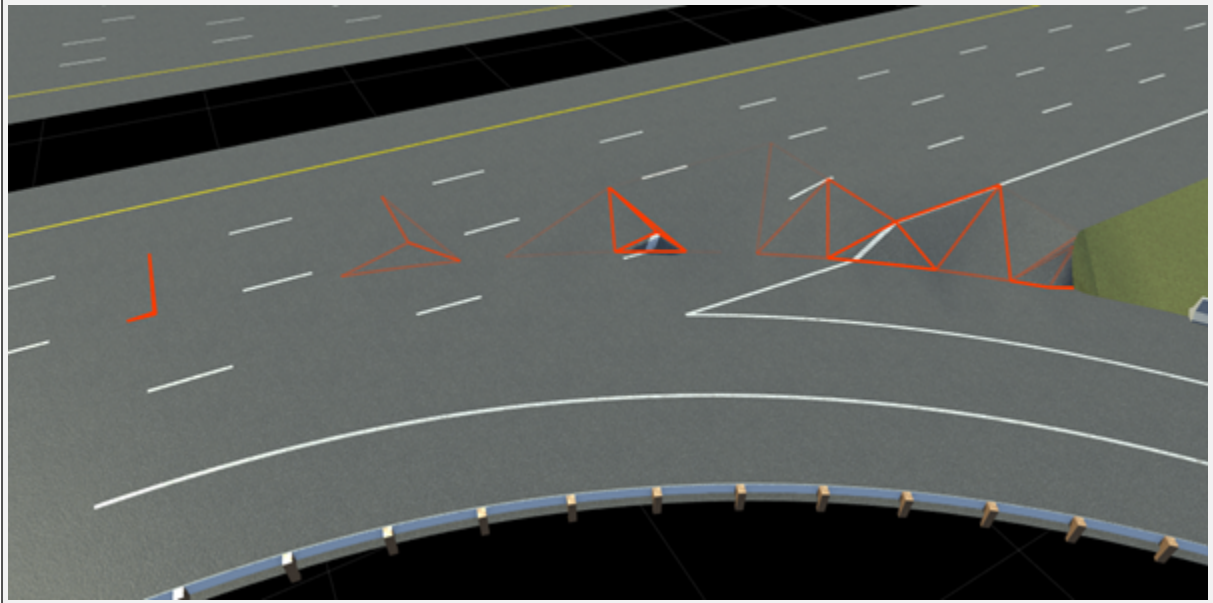
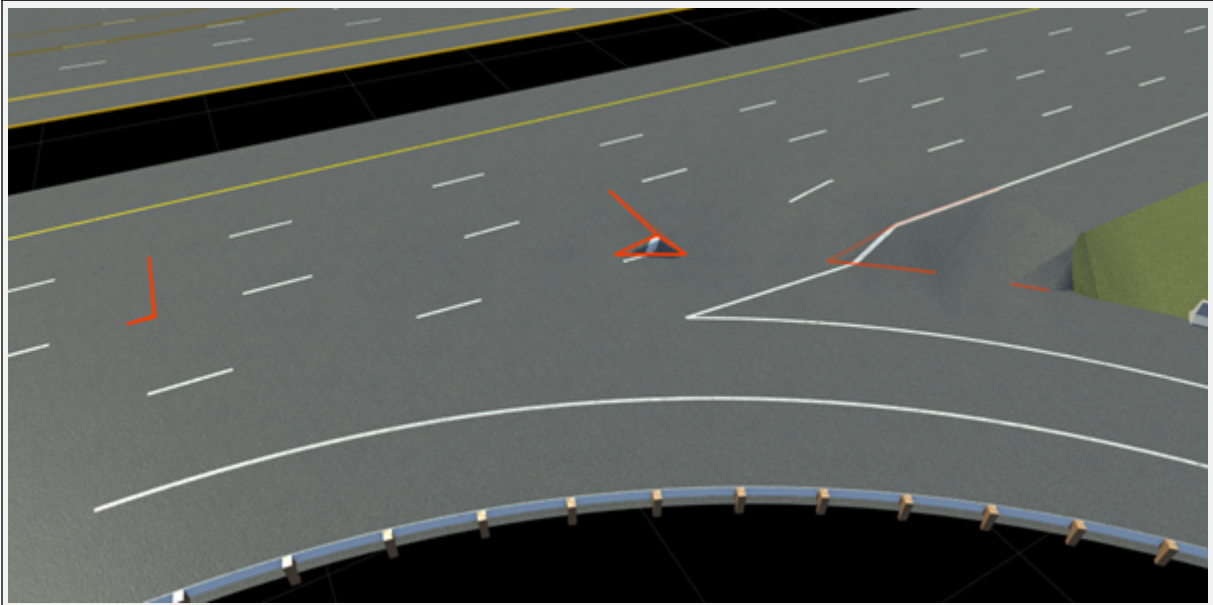
You can enable visualization of geometric issues related to triangulation. To modify geometry issue settings, use the Geometry Issue Settings dialog box under the **Tools** menu.



Angle Threshold

The **Angle Threshold** parameter controls the angle threshold (in degrees) at which geometric issues are displayed. This angle directly translates into the angle between adjacent faces of the road mesh. These images show the same road section with different **Angle Threshold** values used for displaying geometric issues. As the **Angle Threshold** value increases, the number of issues that are detected decreases.

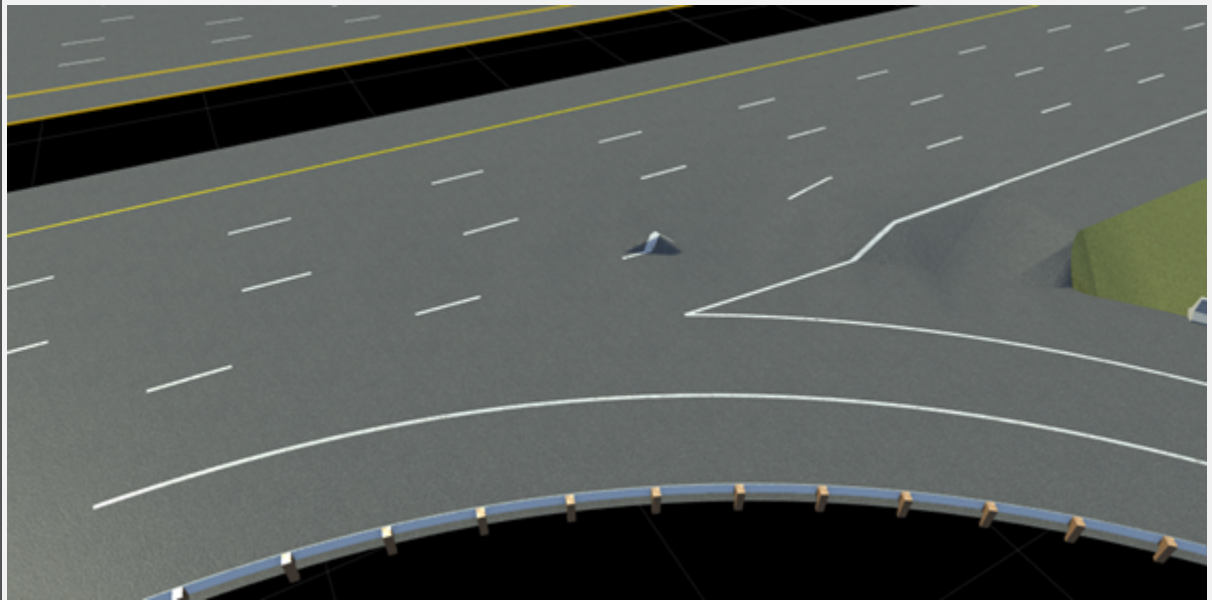


Angle Threshold: 10 degrees**Angle Threshold: 30 degrees**

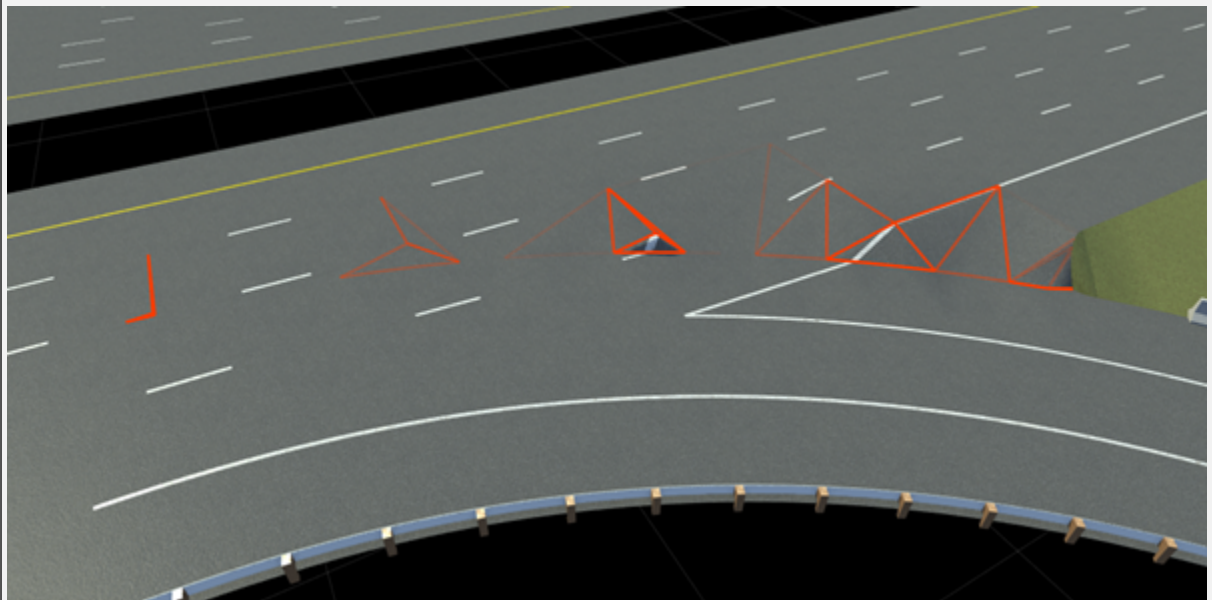
Show Edge Visualization

The **Show Edge Visualization** parameter enables visualization of the geometric issues. This toggle is applied when the dialog box is closed and corresponds directly to the **View > Geometric Issues** menu option.

Edge visualization disabled



Edge visualization enabled



Detect Geometry Issues

Running **Detect Geometry Issues** prints the current state of the geometric issues to its own **Output** pane, and to the RoadRunner standard **Output** pane. Each issue contains its own URL that focuses the camera on each issue.

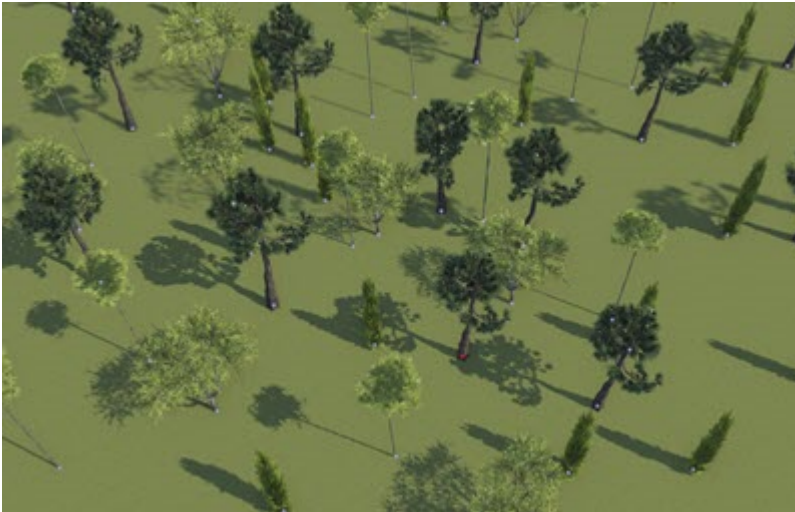
Found Geometry Issues

Output

- Edge with angle 13.6431° at: (-78.9393, 16.5996, 6.73022)
- Edge with angle 13.1922° at: (-70.9352, 14.8873, 6.97454)
- Edge with angle 12.5162° at: (-85.3799, 24.9748, 6.21304)
- Edge with angle 12.0009° at: (-69.0391, 15.9658, 6.71417)
- Edge with angle 11.5995° at: (-70.2454, 16.3665, 6.64437)
- Edge with angle 11.4919° at: (-81.25, 17.4688, 7.2)
- Edge with angle 11.4918° at: (-81.25, 15.7188, 7.2)
- Edge with angle 11.4452° at: (-71.439, 16.8032, 6.57456)
- Edge with angle 11.442° at: (-74.9338, 18.3257, 6.36515)
- Edge with angle 11.0295° at: (-85.008, 24.4595, 6.37663)
- Edge with angle 10.6944° at: (-79.125, 14, 7.2)
- Edge with angle 10.581° at: (-86.539, 22.8016, 7.16943)
- Edge with angle 10.3945° at: (-86.6083, 22.7864, 7.19087)

Close

Point Editing



Some RoadRunner objects, such as prop instances, are modeled as points. This topic provides common steps to create, delete, and modify these point instances.

For general information about selecting and deleting objects, see “Design Scenes”.

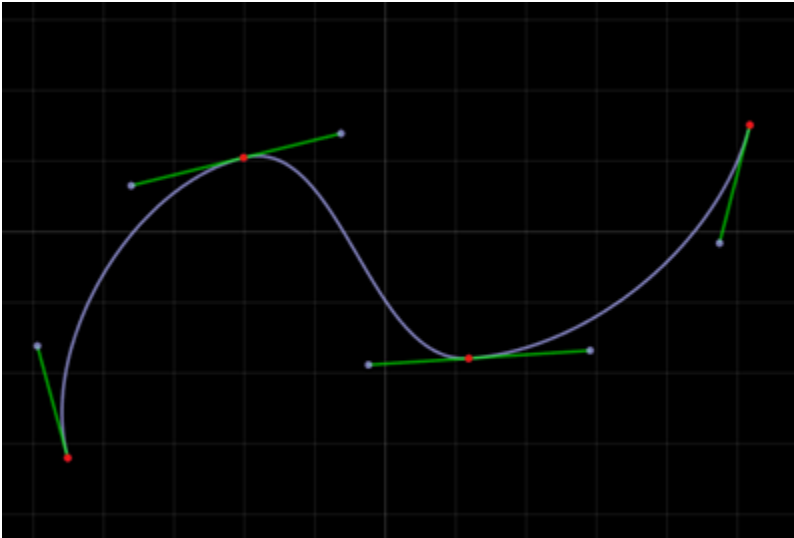
Create a New Point

- 1 Select the point tool that corresponds to the type you want to create (for example, select the **Prop Point Tool** for editing prop point instances).
- 2 Some tools will require an appropriate asset to be selected in the **Library Browser** before a curve can be created (for example, select **Prop Model Assets** if you are creating a prop instance).
- 3 Right-click to create a point. The new point is automatically assigned to the selected asset.
- 4 Optional: Continue holding the mouse button and drag to move the point after initial creation.

Move a Point

- 1 Select the point tool that corresponds to the type you want to modify (for example, select the **Prop Point Tool** for editing prop point instances).
- 2 Click and drag a point to move it to a new location.

Curve Editing



Some RoadRunner data models are built on top of curve sequences, including roads, prop curves, and marking curves. This topic provides common steps to create, delete, and modify these curve instances.

For general information about selecting and deleting objects, see “Design Scenes”.

Create a New Curve

- 1 Select the curve tool that corresponds to the type you want to create (for example, select the **Marking Curve Tool** if you want to build a marking curve).
- 2 Some tools will require an appropriate asset to be selected in the **Library Browser** before a curve can be created (for example, select **Lane Marking Assets** to create marking curves).
- 3 Ensure that no objects are selected (for example, by selecting **Edit** and then **Deselect All** in the menu bar).
- 4 Right-click (and optionally drag) to create a curve with a single starting point. The new curve will automatically be assigned the selected asset.
- 5 Right-click (and optionally drag) to extend the curve by adding additional control points.

Extend a Curve at Its Ends by Adding Control Points

- 1 Select the curve tool that corresponds to the type you want to modify (for example, select the **Marking Curve Tool** to build a marking curve).
- 2 Select the curve you want to edit.
- 3 Click a control point at the end that you want to extend.
- 4 Right-click (and optionally drag) to add an additional control point.

Add Control Points to the Interior of a Curve

- 1** Select the curve tool that corresponds to the type you want to modify (for example, select the **Marking Curve Tool** to build a marking curve).
- 2** Select the curve you want to edit.
- 3** Right-click (and optionally drag) the curve at the location where you want to insert a new control point.

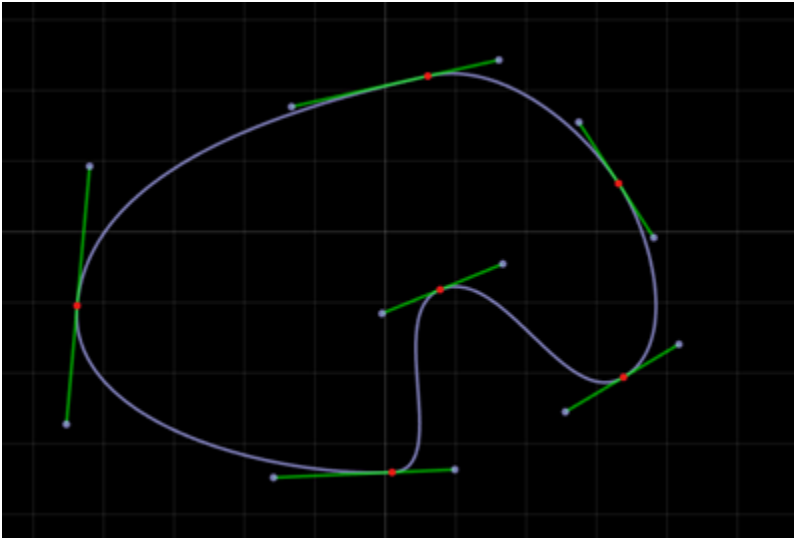
Move a Control Point

- 1** Select the curve tool that corresponds to the type you want to modify (for example, select the **Marking Curve Tool** to build a marking curve).
- 2** Select the curve you want to edit.
- 3** Click and drag a control point to move it to a new location.

Change the Tangents of a Curve

See “Tangent Editing” on page 2-57.

Polygon Editing



Some RoadRunner data models are polygon-based, such as prop and marking polygons. This topic provides common steps to create, delete, and modify these polygon instances.

For general information about selecting and deleting objects, see “Design Scenes”.

Create a New Polygon

- 1 Select the polygon tool that corresponds to the type you want to create (for example, select the **Marking Polygon Tool** to build a marking polygon).
- 2 Some tools require an appropriate asset to be selected in the **Library Browser** before a polygon can be created (for example, select **Polygon Marking Assets** if you are creating marking polygons).
- 3 Ensure that no objects are selected (for example, by selecting **Edit** and then **Deselect All** from the menu bar).
- 4 Right-click (and optionally drag) to create a polygon with a single starting point. The new polygon will automatically be assigned the selected asset.
- 5 Right-click (and optionally drag) to extend the polygon and add additional control points.

Add Control Points to a Polygon

There are two ways to add control points to a polygon.

Insert a Control Point Next to an Existing Point

- 1 Select the polygon tool that corresponds to the type you want to modify (for example, select the **Marking Polygon Tool** to build a marking polygon).
- 2 Select the polygon you want to edit.
- 3 Select a control point next to the point you want to add.
- 4 Right-click (and optionally drag) to add additional control points.

Insert Control Points by Splitting a Polygon Edge

- 1** Select the polygon tool that corresponds to the type you want to modify (for example, select the **Marking Polygon Tool** if you want to build a marking polygon).
- 2** Select the polygon you want to edit.
- 3** Right-click (and optionally drag) the polygon edge where you want to insert a new control point.

Move a Control Point

- 1** Select the polygon tool that corresponds to the type you want to modify (for example, select the **Marking Polygon Tool** to build a marking polygon).
- 2** Select the polygon you want to edit.
- 3** Click and drag a control point to move it to a new location.

Change the Tangents of a Polygon

See “Tangent Editing” on page 2-57.

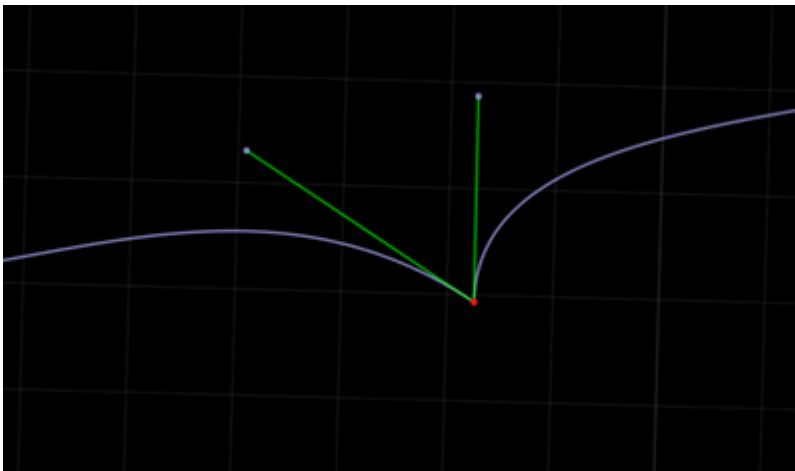
Tangent Editing



Some RoadRunner data models are built on top of curves and curve sequences, including roads, prop curves and polygons, marking curves and polygons, and the terrain surface graph. The control points of these curves contain tangents that can be adjusted to smooth or kink the resulting boundaries. This topic provides common steps for editing tangents and enforcing tangent continuity.

Adjust a Tangent

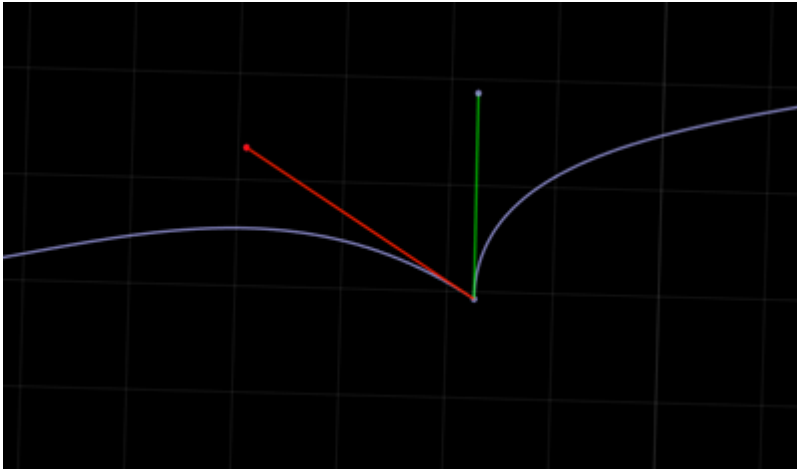
- 1 Click the parent object to expose the tangent views.



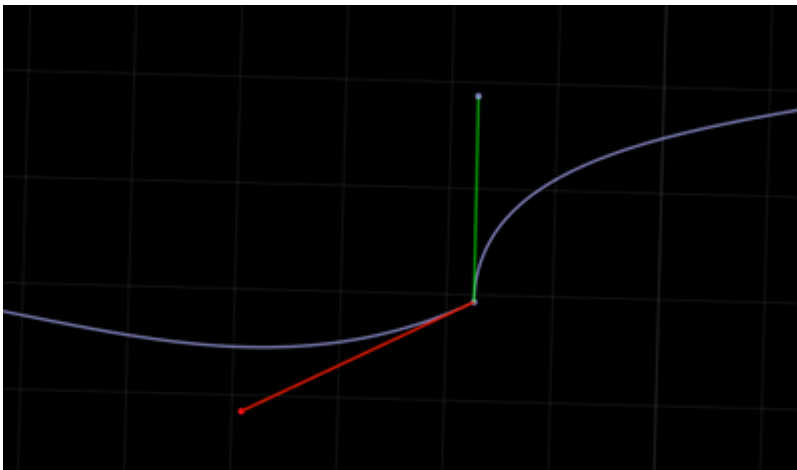
Note For some data types, such as road height profiles, the tangents are exposed once the parent is selected.

For other data types, such as curves or polygons, this might require clicking the parent object first to expose the control points.

- 2 Click the end point of the tangent handle.



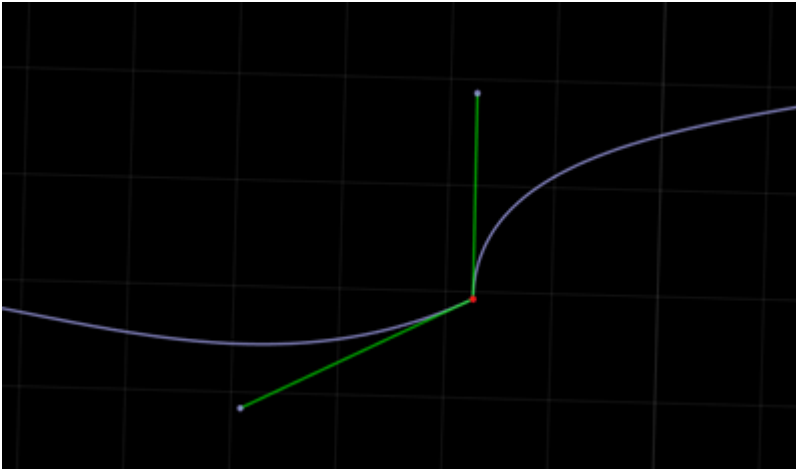
- 3 Click and drag to set the direction and scale of the tangent.



Make Tangents Continuous

To automatically enforce continuity, use the **Connect Tangents** operation:

- 1 Click the parent object to expose the tangent views.



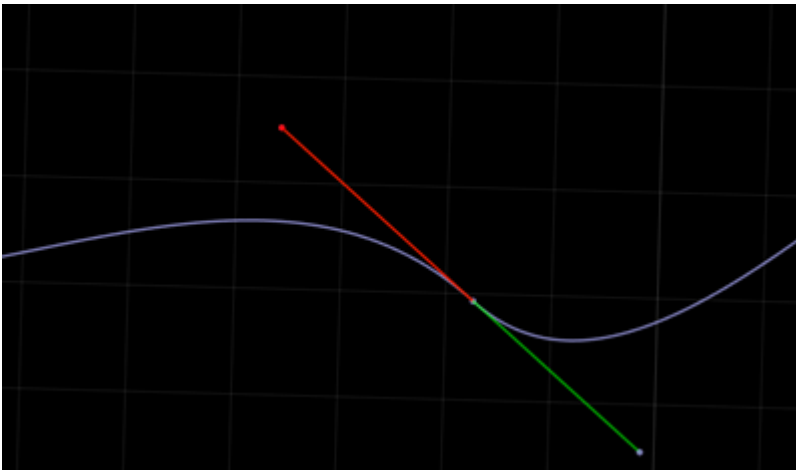
Note For some data types, such as road height profiles, the tangents are exposed once the parent is selected.

For other data types, such as curves or polygons, this might require clicking on the parent object first to expose the control points.

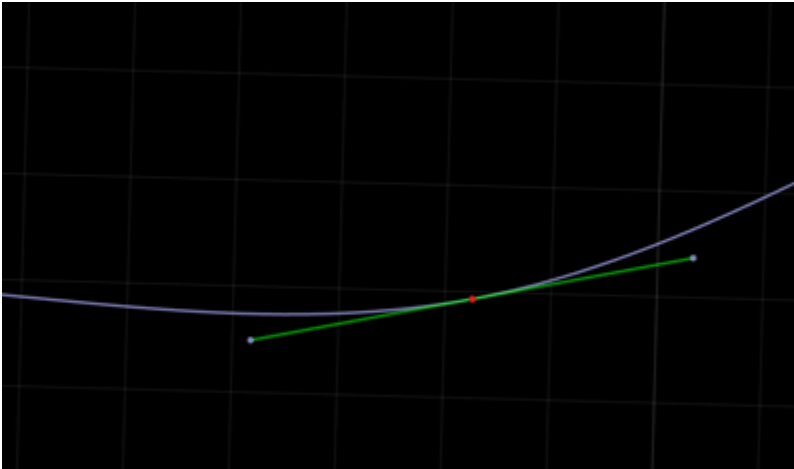
2



Click the **Connect Tangents** button.



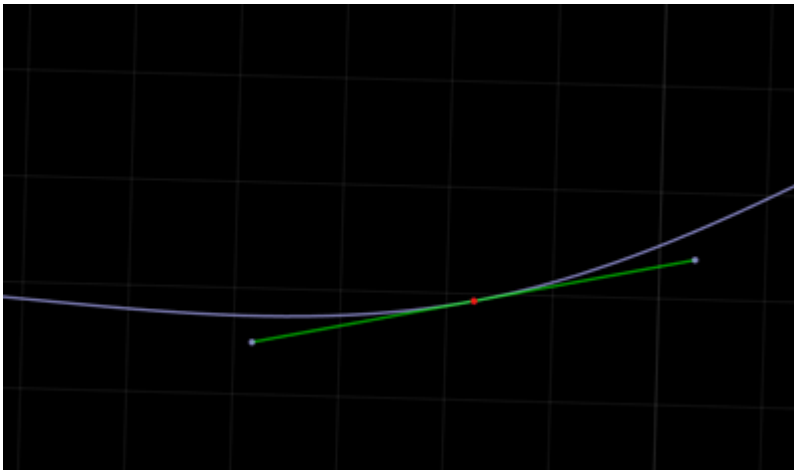
Tangents will now be enforced, even through additional edits.



Make Tangents Discontinuous

To remove automatic continuity constraints, use the **Disconnect Tangents** operation:

- 1 Click the parent object to expose the tangent views.



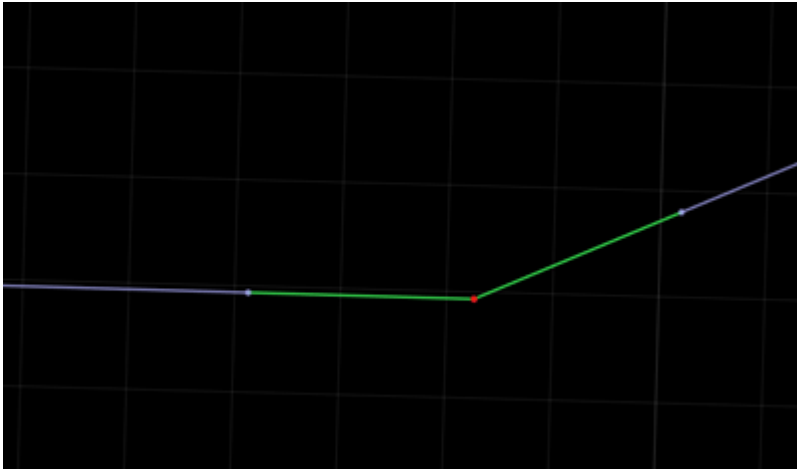
Note For some data types, such as road height profiles, the tangents are exposed once the parent is selected.

For others, such as curve or polygons, this might require clicking the parent object first to expose the control points.

- 2



Click the **Disconnect Tangents** button.

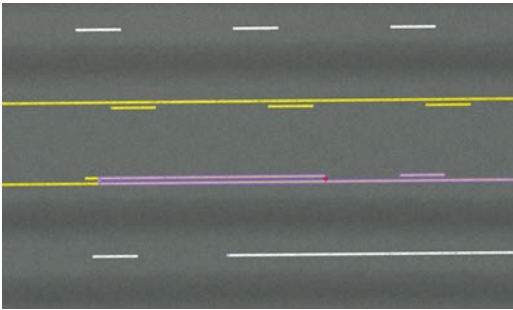


See Also

Related Examples

- “Choose a RoadRunner Tool” on page 2-27

Span Editing



Various attributes are represented as parametric spans along lanes, roads, and other objects. This topic provides common steps to create, delete, and modify these span instances. Various tools use span editing concepts, such as the **Lane Marking Tool**, **Prop Span Tool**, or **Road Construction Tool**.

Span Overview

Span-based attributes are defined by the following components.

Parent Object

Span-based attributes are defined parametrically along a curve-based parent object. Typically, the parent object is either a road (as in the **Road Construction Tool**) or a lane (as in the **Lane Marking Tool**).

Span Nodes



Span nodes (red circles) selected in the **Lane Marking Tool**. These nodes indicate locations where the marking type changes along the lane.



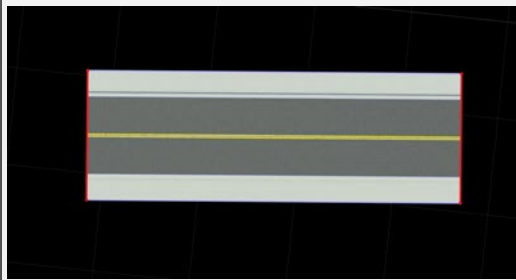
Span nodes (red lines) selected in the **Road Construction Tool**. These nodes indicate where bridges start and end along a road.

Span nodes are parametric objects along a parent curve that define where attribution changes. The visual representation of nodes differs depending on the tool (as shown in the previous images).

Span nodes can be moved along the parent curve. Nodes can also be added along a curve, and existing nodes can be deleted.

Node locations are automatically updated when the parent curve is modified (for example, when the parent road's shape is changed).

Span End Nodes



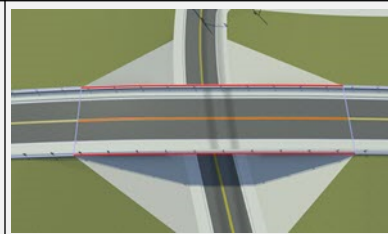
Span end nodes selected in the **Road Construction Tool**.

Span end nodes are a special type of span node that lie at the beginning and end of the parent curve. For most types of span-based attributes, these span end nodes are automatically created and cannot be deleted.

Spans



Span selected in the **Lane Marking Tool**. This span defines the lane marking type between two nodes.



Span selected in the **Road Construction Tool**. This span defines the construction type (for example, bridge) between two nodes.

A span is a range along a parent curve bounded by two span nodes. For most types of span-based attributes, the span is automatically created between the span end nodes and cannot be deleted.

Select a Span or Span Node

The steps to select a span differ slightly depending on the tool, but the steps are typically similar to the following.

- 1 Select the parent object containing the span (typically either a road or a lane on a road).
- 2 Select the desired span or span node.

Create a New Span Node

New span nodes are created by splitting a span into two spans.

- 1 Select the parent object containing the span.
- 2 Right-click an existing span at the location where you want to insert the new node.

Note In most cases, any attributes stored in the span are copied into the two new spans.

Edit Attributes of a Span or Span Node

- 1 Select a span or span node.
- 2 Adjust the properties in the **Attributes** pane.

Alternatively, for asset-based attributes such as in the **Lane Marking Tool** and **Prop Span Tool**, click and drag a compatible asset type from the **Library Browser** to the span or span node.

Note Some span-based attributes only store data on the spans, others only store data on the span nodes, and some store data on both.

Span nodes always have a "Distance" attribute that defines the distance of the node along the parent curve.

Move a Span Node

- 1 Select the parent object containing the span.
- 2 Click and drag the node along the parent curve.

Alternatively:

- 1 Select the parent object containing the span.
- 2 Select the span node.
- 3 In the **Attributes** pane, adjust the **Distance** attribute.

Note Most span nodes cannot be moved past another node, and must remain a minimum distance from surrounding nodes.

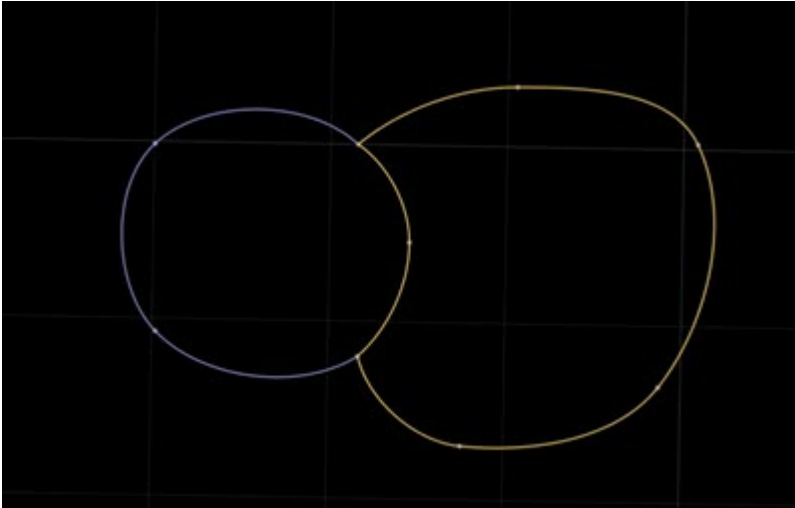
Delete a Span Node

- 1 Select the parent object containing the span.
- 2 Select the span node.
- 3 Delete the span node.

Tips for Deleting Nodes

- In most cases, the span end nodes (the nodes at the end of the parent curve) cannot be deleted.
- Deleting a span node combines the two attached spans into a single span. In most cases, the single span receives the attributes of the longer span. The shorter span is removed.

Region Graph Editing



Some RoadRunner data models are built on top of graphs of curve-bounded regions. This topic provides common steps to create, delete, and modify these region graphs.

Many of the editing concepts for region graphs are similar to the concepts for “Design Scenes”. For example, you can create, edit, and delete curve-based graph edges that behave like most curves in RoadRunner.

Region graphs differ in two regards:

- Edge Connectivity — Graph edge curves can be connected end-to-end.
- Regions — Whenever a closed loop of graph edges is formed, a region is created in the interior.

Refer to the “Design Scenes” page for general information about selecting and deleting objects.

Create a Graph Edge Curve

- 1 Select the graph region tool that corresponds to the type you want to create (for example, select the **Surface Tool** if you want to edit surfaces).
- 2 If you want to start the new edge at an existing node, select the existing node. Otherwise:
 - a Ensure that no objects are selected (for example, by using the **Edit > > Deselect All** option in the menu bar).
 - b Right-click (and optionally drag) to create an initial graph node.
- 3 Optional: Move the pointer over an existing node if you want to end the curve at that node.
- 4 Right-click (and optionally drag) to create a second graph node and a graph edge curve in between.

Split a Graph Edge Curve

- 1 Select the graph region tool that corresponds to the type you want to create (for example, select the **Surface Tool** to edit surfaces).

- 2 Right-click a graph edge curve to split it into two curves.

Move a Graph Node

- 1 Select the graph region tool that corresponds to the type you want to create (for example, select the **Surface Tool** to edit surfaces).
- 2 Click and drag the graph node you want to move.

Change the Tangents of a Graph Edge Curve

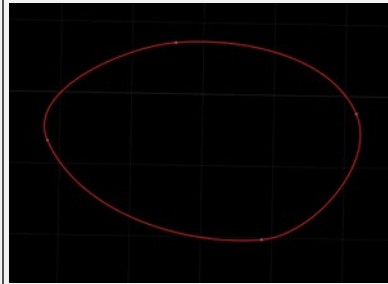
See "Tangent Editing" on page 2-57.

Create a Region

Regions are automatically created whenever a closed loop of graph edges is formed:



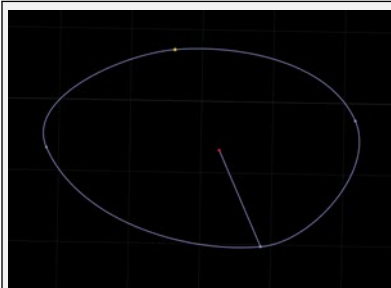
Sequence of connected graph edge curves (no region)



Region is automatically created when the open ends are connected.

Split a Region

Each closed loop of graph edges automatically forms a region, so one region can be split into two by forming a path of graph edge curves between two of the points on the region exterior:



Single region



Region is automatically split into two regions when two closed loops are formed.

Regions With Holes

Only the **Surface Tool** fully supports holes.

Graphics and Startup Issues

System Requirements

RoadRunner is primarily a 3D graphics application. It requires a graphics card with support for OpenGL version 3.2 or higher.

Check that your system meets the minimum RoadRunner System Requirements on page 1-3. RoadRunner might still work with some lower specification system version, but you might experience poor performance.

Graphics Drivers

If you are experiencing rendering issues or crashes on startup, check to make sure that you are running the latest graphics drivers for your system. Some computer manufacturers install custom or unstable drivers. Windows Update has also been known to automatically install problematic graphics drivers.

Downloading drivers directly from your graphics card manufacturer is recommended. Common graphics card manufacturers include:

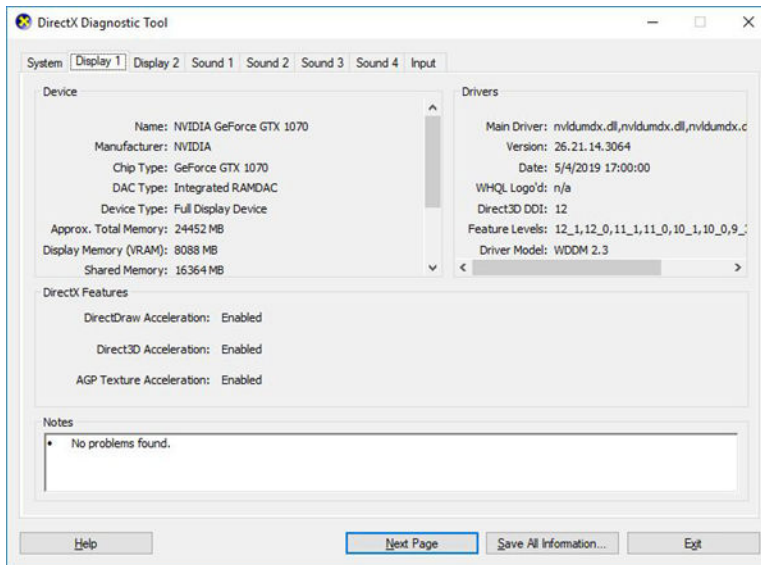
- NVIDIA (drivers download page)
- AMD (drivers download page)
- Intel (drivers download page)

On Linux, if your graphics card manufacturer does not supply drivers for Linux, try updating your Mesa graphics drivers.

If you are unsure which type of graphics card you have, you can often determine that by visiting your computer manufacturer's web page and searching for drivers for your system. This search usually involves entering a model (or serial) number or installing an autodetection application.

On Windows, you can determine which graphics cards you have installed by following these steps:

- 1** Click the **Start** button and type Run.
- 2** Type `dxdiag` and press **Enter**.
- 3** Inspect the **Name** and **Manufacturer** on the **Display** tabs. You might have multiple graphics cards installed, so inspect each listed **Display** tab.



Laptops

RoadRunner can run on laptops with sufficiently powerful graphics cards. Newer laptops with low-powered graphics cards (such as Intel embedded graphics chips) might still be able to run RoadRunner acceptably.

To conserve battery life, modern laptops often have two graphics cards, for example, higher-powered NVIDIA graphics combined with lower-powered Intel embedded graphics. In these cases, RoadRunner requests to use the higher-power graphics card, but there is no guarantee that your system will obey that request.

If your laptop has multiple graphics cards (often true if your laptop advertises an NVIDIA or AMD graphics cards), then it is recommended that you check that your system provides the higher-power graphics card to RoadRunner.

The steps for this process differ depending on the laptop manufacturer, graphics driver, and other factors. For additional help, see these links:

- <https://www.techadvisor.co.uk/how-to/pc-components/how-set-default-graphics-card-3612668/>
- [https://www.nvidia.com/content/Control-Panel-Help/vLatest/en-us/mergedProjects/nv3d/Manage_3D_Settings_\(reference\).htm](https://www.nvidia.com/content/Control-Panel-Help/vLatest/en-us/mergedProjects/nv3d/Manage_3D_Settings_(reference).htm) (see "Program Settings" section; for RoadRunner, select the **high-performance NVIDIA processor** option)

Remote Desktops

If you are using RoadRunner while connected to a remote desktop, you might encounter performance issues. To resolve these issues, consider using one of these options.

NVIDIA Remote Desktop Acceleration

If you are using Microsoft® Remote Desktop on newer NVIDIA GeForce drivers, follow the procedure described in the "Accelerate Windows Remote Desktop" section of <https://developer.nvidia.com/designworks>.

Chrome Remote Desktop

Chrome™ Remote Desktop from Google® provides a solution for remote access that supports newer OpenGL applications. For details, see <https://remotedesktop.google.com/>.

Video Card Connection

On desktops, you might also want to check that your monitor is connected to a video card, rather than to a low-powered graphics card (also called "onboard" or "integrated" graphics). Trace your monitor cable to make sure that it is not plugged into the built-in graphics port, but rather into a video card in one of the expansion slots.

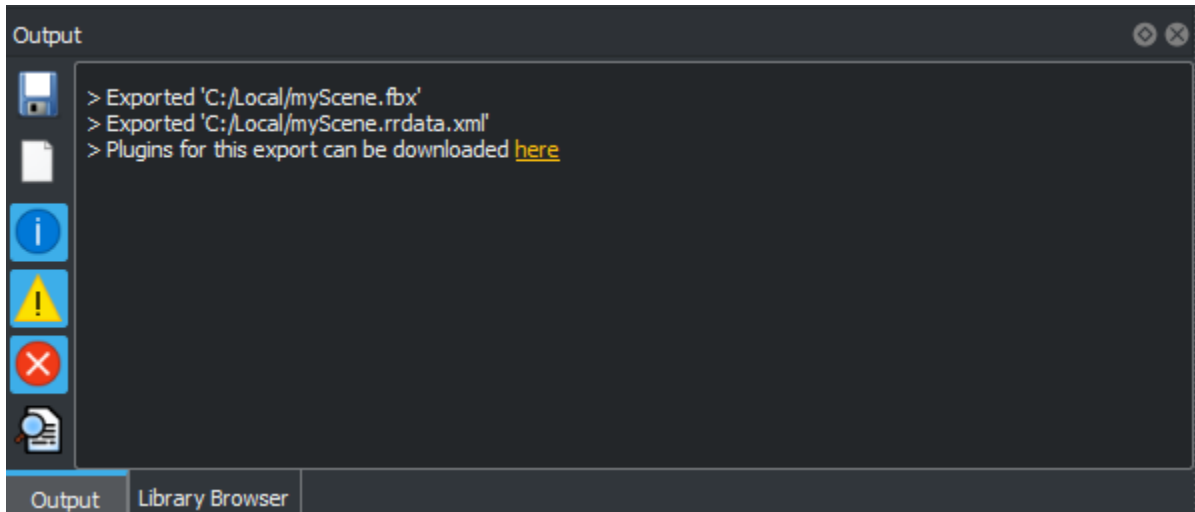


Further Support

If the previous information does not resolve your issue, report your issue to MathWorks Technical Support. It is helpful to include your log files. For more details, see “Obtain RoadRunner Log Files” on page 2-72.

Obtain RoadRunner Log Files

While debugging RoadRunner issues, MathWorks Technical Support might request your RoadRunner log files. Log files are all messages, warnings, and errors related to RoadRunner that are printed to the **Output** pane. This image shows a sample **Output** pane with logged information.



This information can be helpful for debugging certain problems, especially problems related to loading asset files.

Locate Log Folder

To obtain the log from the current RoadRunner session, from the RoadRunner menu, select **Tools > Debug > Open Log Folder**

To locate the log folder that contains log files for all RoadRunner sessions, in Windows, click **Start**, and then type:

```
%appdata%\MathWorks\RoadRunner\<release version>\Logs
```

To locate log files in Linux, navigate to this folder:

```
~/local/share/MathWorks/RoadRunner/<release version>/Logs
```

Provide Log File Contents to MathWorks Technical Support

- 1 Locate the log folder (see previous section).
- 2 Zip (or tar) the contents of the log folder.
- 3 Attach the zip file to a new or existing ticket for MathWorks Technical Support.

Import Data

- “Importing OpenDRIVE Files” on page 3-2
- “Decompress LAZ Files” on page 3-4
- “GIS Data Resources for RoadRunner” on page 3-7

Importing OpenDRIVE Files

RoadRunner can import OpenDRIVE files and convert the data to the internal road format.

OpenDRIVE Overview

RoadRunner can visualize and import OpenDRIVE 1.4 data, converting the data to the internal road format during import. The OpenDRIVE data is visualized before import. The import option is designed to enable editing of OpenDRIVE files.

Import OpenDRIVE into Scene

- 1 Add the OpenDRIVE file to the **Library Browser** in RoadRunner.
- 2 Click and drag the OpenDRIVE file from the **Library Browser** into the scene. This action switches to the **OpenDRIVE Viewer Tool**.
- 3 Click the imported file in the 3D scene and click the **Build Scene** button in the toolbar on the left.

Import Objects, Signals, and Markings

The OpenDRIVE importer uses a configuration XML file to map OpenDRIVE <object>, <signal>, and <marking> entries to RoadRunner assets. This configuration file is also used to define the correlation during export. For more details, see the "OpenDRIVE User Asset Configuration" section of "Export to OpenDRIVE" on page 5-15.

Import Options

The OpenDRIVE import dialog box provides these options:

- Import Props: If this option is enabled, all <object> entries are mapped to props or markings as needed.
- Import Signals: If this option is enabled, all <signal> entries are mapped to signals or signs as needed.

Limitations

- Lane connections are stored differently between OpenDRIVE and RoadRunner. Lanes that were connected in OpenDRIVE might be disconnected in RoadRunner. A validation error is printed if this is the case.
- Some OpenDRIVE files do not distinguish between logical and physical position of signs and signals. During import, signs and signals might be placed directly on the road surface at the stopping location.
- RoadRunner road plan geometry requires that segments (line, arc, clothoid, or parametric cubic) be continuous. Some OpenDRIVE data might be defined as a series of <line> segments that are mapped to a continuous curve during import. The OpenDRIVE geometry type <poly3> is not supported.
- RoadRunner roads cannot connect to themselves. Connections in OpenDRIVE of a road connected to itself is ignored during import.

- RoadRunner creates junctions based on the overlap of roads. If roads are overlapping in OpenDRIVE, a junction might be created during import.

Decompress LAZ Files

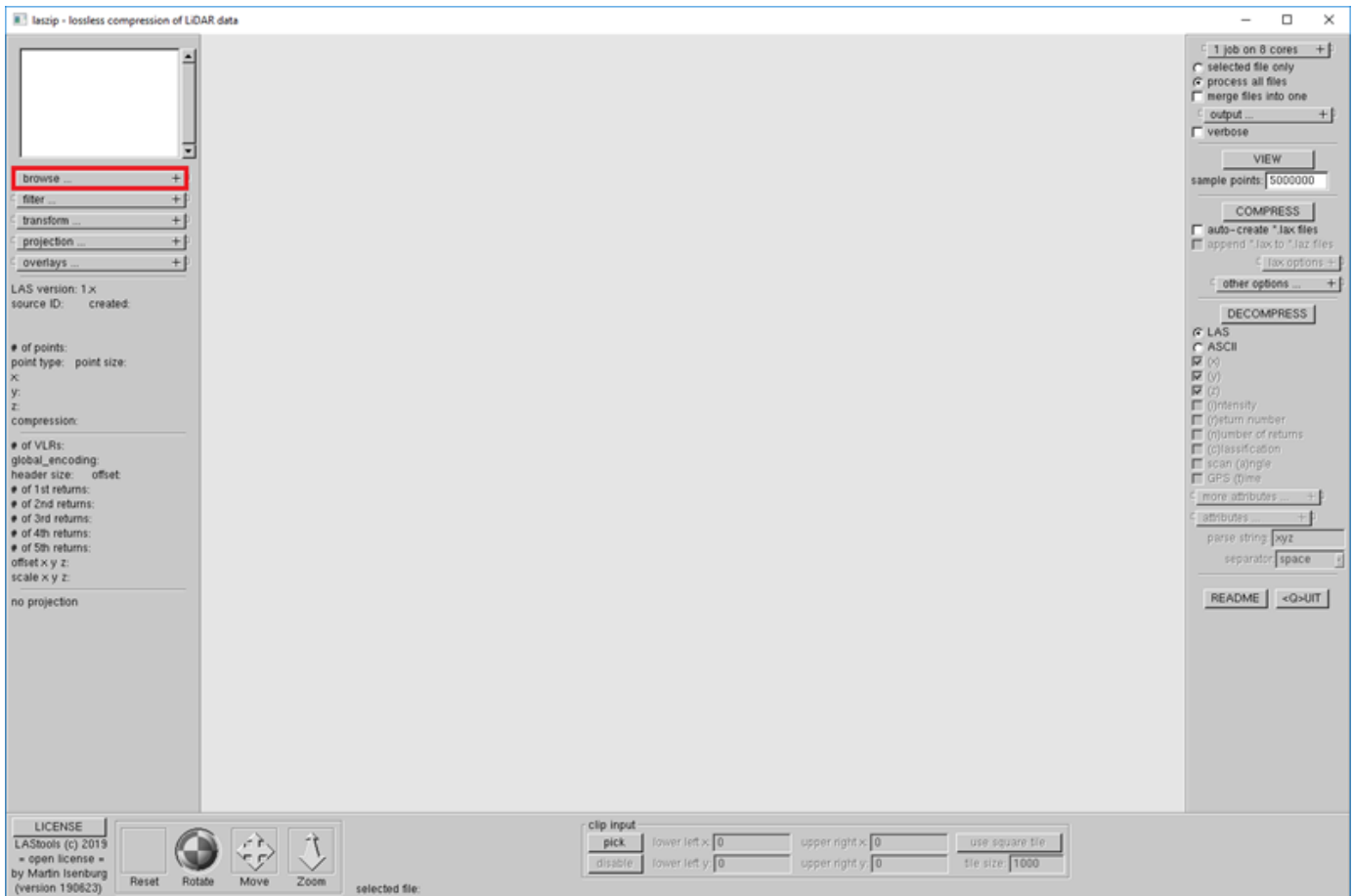
RoadRunner software does not support some LAZ files, resulting in this error: "The LAZ schema is not recognized". To resolve this issue, you can decompress the LAZ file into an LAS file.

Decompression Process

- 1 To get started, on Windows, get the latest version of LASzip (found here). To get started on Linux, build an executable for your operating system.
- 2 For both operating systems, run the LASzip executable.

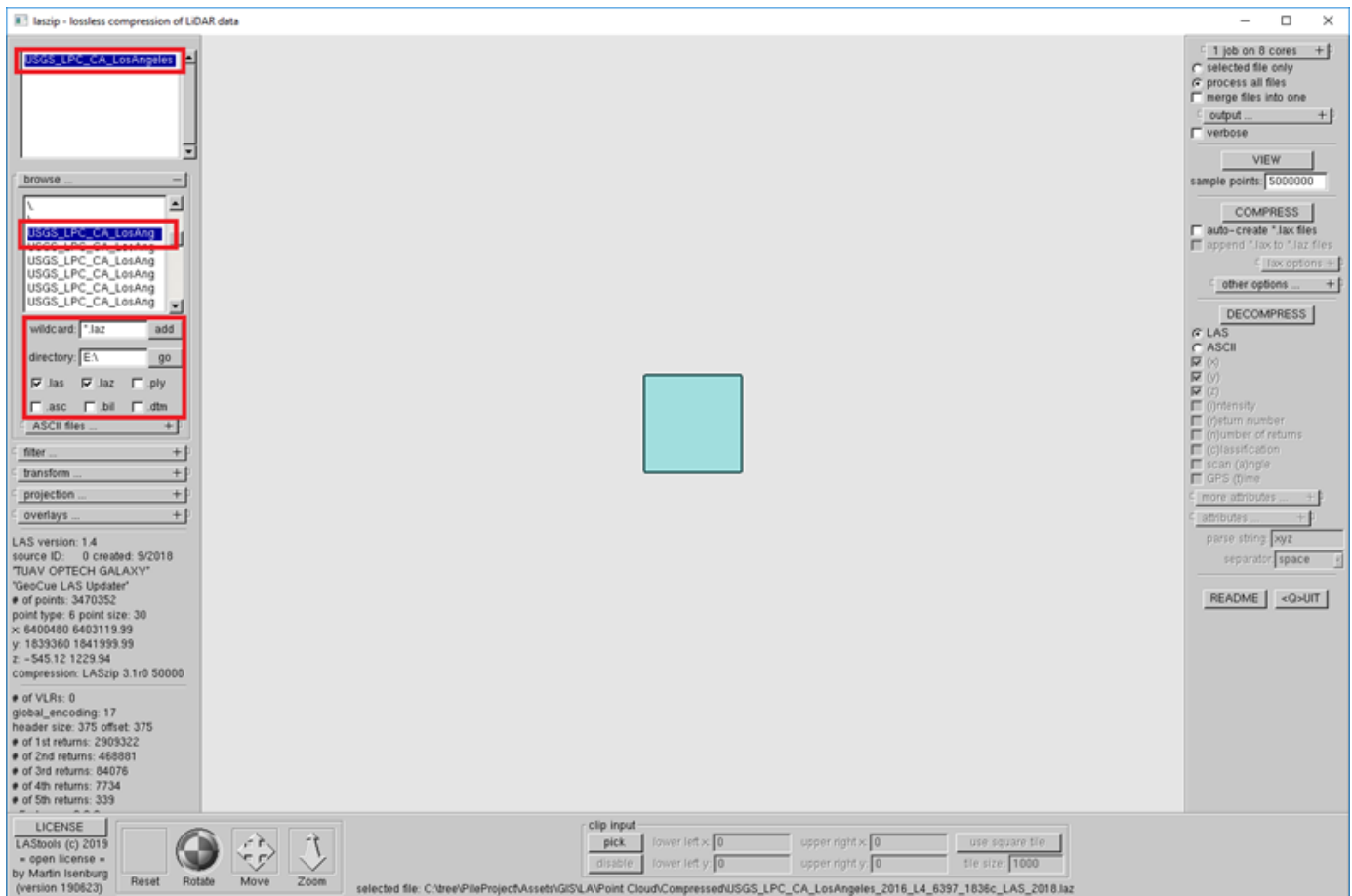
Tip LASzip can sometimes have strange behavior when clicking or selecting in the interface. To fix this issue, try maximizing the LASzip window or increasing the window size.

- 3 Click **browse**.



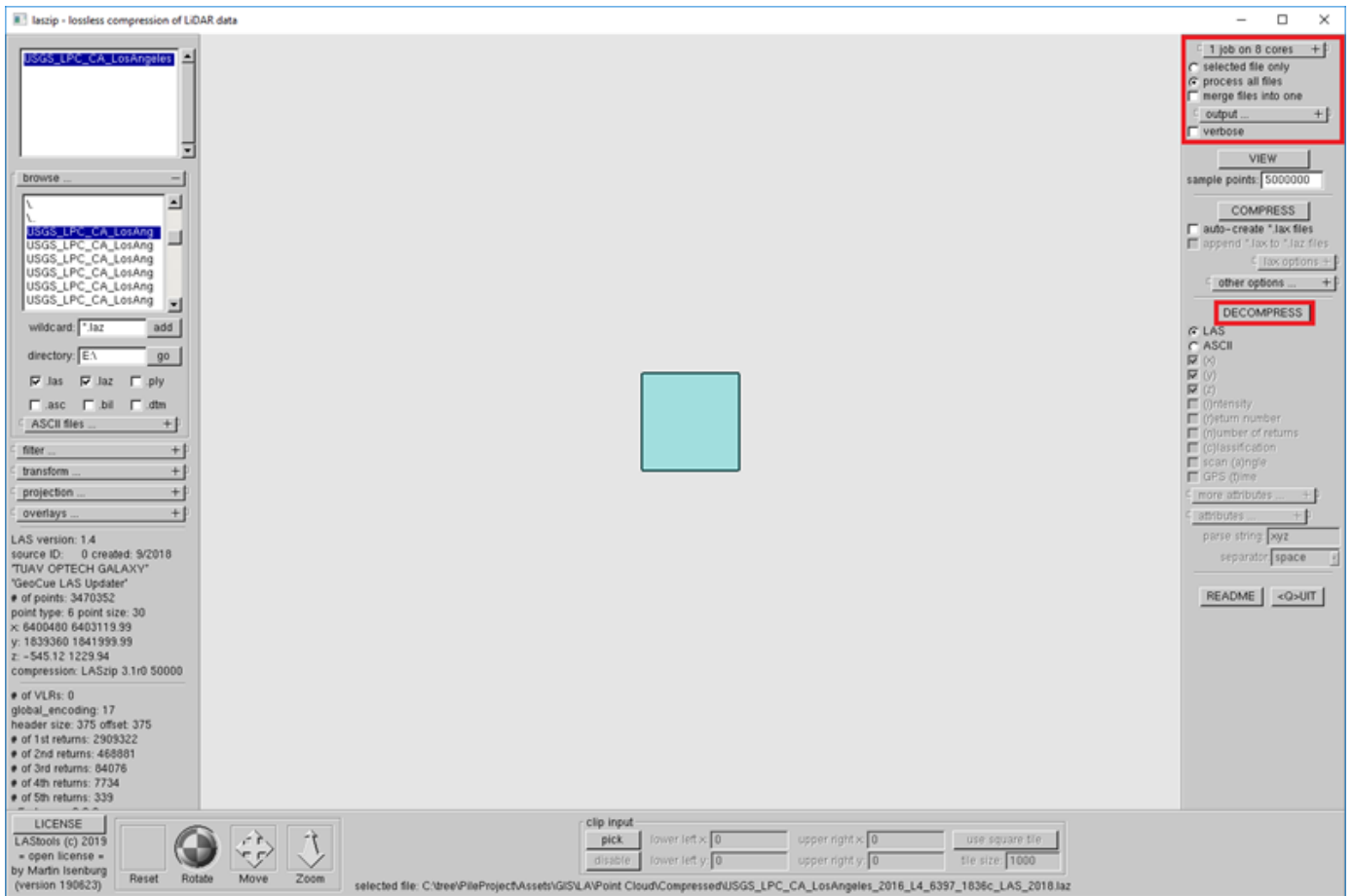
- 4 Find the desired LAZ file on your system. You can go to a specific directory using the **directory** field and clicking **go**.

Tip To add multiple LAZ files, you can use the **wildcard** field to specify which types of files to add. Then click **add**. This action adds all the files fitting that wildcard in the currently browsed directory.

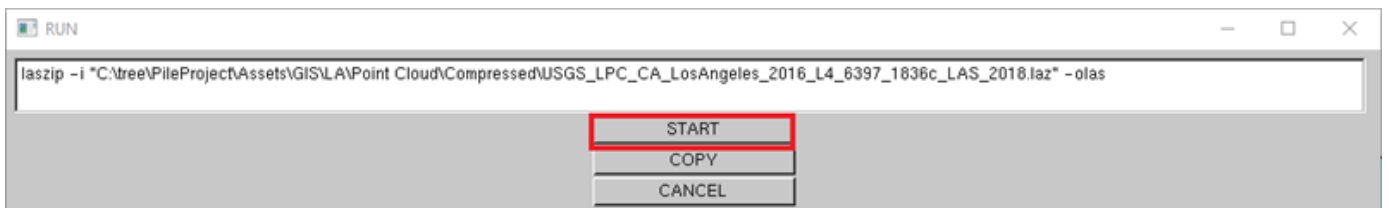


- Adjust settings for output and processor usage in the upper-right corner of the window. Then click **Decompress**.

3 Import Data



7 Click **Start** to run the decompression.



8 Once the RUN window closes, the decompression is complete. Assuming default settings, you can find your LAS files beside your LAZ files in the same directory.

GIS Data Resources for RoadRunner

Importing GIS Data in RoadRunner

After you obtain geographic information system (GIS) data, the process for importing the data into RoadRunner is the same as importing any other data, regardless of where the data was obtained.

RoadRunner supports a multitude of different formats. Refer to the documentation for lists of supported formats for each type of GIS data. The most common file formats are:

- Raster data (satellite imagery and elevation), such as GeoTIFF and JPEG 2000
- Lidar data, such as LAZ and LAS

USGS

The U.S. Geological Survey (USGS) provides freely available GIS data¹ for much of the United States. Coverage and quality varies depending on the data type and location.

The table shows USGS interfaces from which you can access GIS data.

| Interface Link | Interface Description |
|---|---|
| https://viewer.nationalmap.gov/basic/ | Basic National Map explorer interface (preferred) |
| https://viewer.nationalmap.gov/advanced-viewer/ | Advanced National Map explorer interface |
| https://earthexplorer.usgs.gov/ | Earth Explorer interface |

Each interface has a different user interface (UI) for selecting locations and data sets, and the different interfaces can contain different data sets. When finding data for a specific project, checking all the USGS interfaces is recommended.

For example, to download data from the Basic National Map explorer interface:

- 1 Find your area of interest by using the interface on the right.
- 2 Select one or more data sources on the left. The "Show Availability" links display the coverage of a given data source.
 - For elevation data, select "Elevation Products" (a 1/9 arc-second is recommended, but that option is not available in all locations).
 - For lidar data, select "Elevation Source Data" (lidar point cloud (LPC) option).
 - For imagery data, select "Imagery - NAIP Plus."
- 3 Click "Find Products", and then click the "results" links to view the download links.
- 4 Use the download links to obtain data. Some data might require downloading multiple tiles to cover your area of interest.

1. Credit: U.S. Geological Survey

Design Scenes

- “Resolve Triangulation Issues in Junctions” on page 4-2
- “How Surfaces Work in RoadRunner” on page 4-4

Resolve Triangulation Issues in Junctions

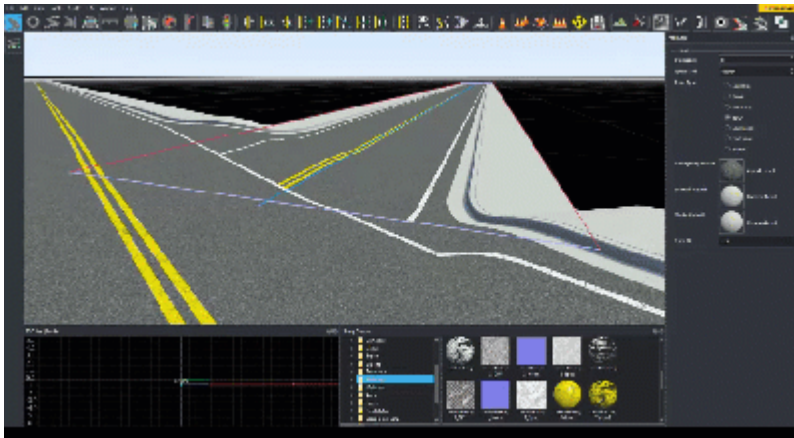
In RoadRunner, a junction represents the complex intersection of multiple roadways, defining a space where multiple surfaces compete for influence over the junction's final surface representation. Even in simple intersections, roads can vary by width, length, bank angle, and elevation. Roads also vary by features such as medians, curbs, and sidewalks, which need to be gracefully clipped from the final result. The goal of RoadRunner software is to unify these overlapping regions into a single representation suitable for simulation use cases. This unification often requires triangulating the junction surface to export into various formats. This task is nontrivial and can often lead to undesirable artifacts in the final junction triangulation.

To avoid common triangulation issues, use these tips.

Adjust Road Elevations

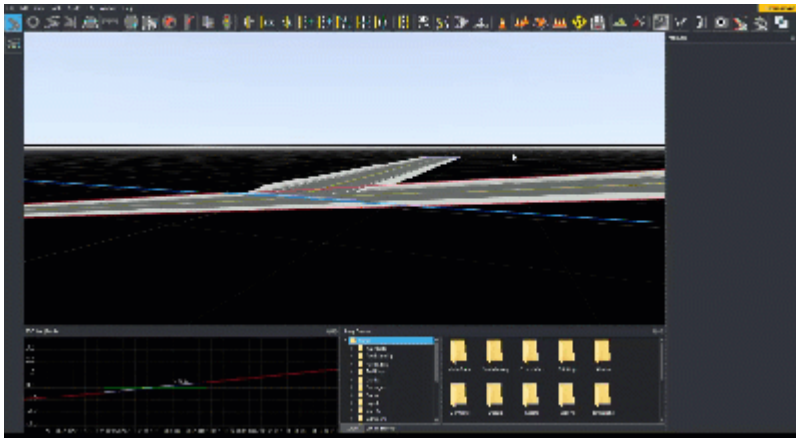
RoadRunner continually detects overlaps with neighboring roads and automatically creates junctions for any roads which overlap within 2 vertical meters. However, given that each road is fully independent, it is possible to create intersections that vary in grade, which can cause undesirable triangulation artifacts. One way to resolve this issue is to adjust road elevations to match as closely as possible within the junction.

From within the **Road Plan Tool**, the RoadRunner 2D Profile editor displays all overlapping roads for any selected road. You can use the tool to raise or lower any road to match the height of other roads by selecting and dragging either the height profile nodes or spans. Dragging a span is equivalent to dragging the nodes on either end.



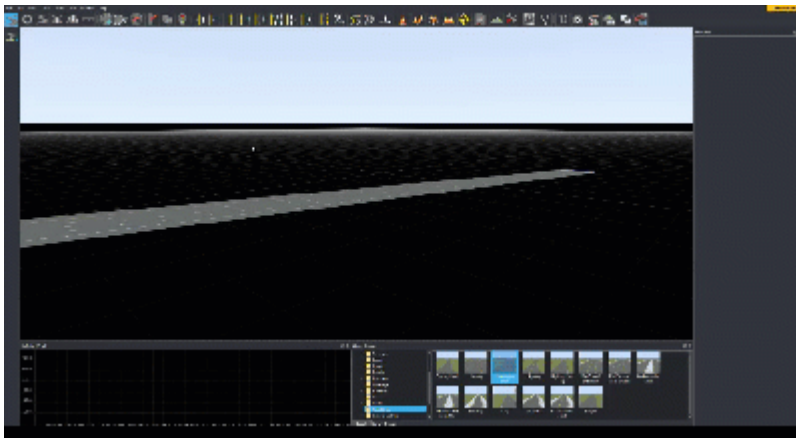
Bank Roads

When two or more roads that intersect have different slopes, the intersections might need to be banked to better align the road surfaces. The RoadRunner **Cross Section Tool** offers an interface to adjust road bank at lane boundary locations. To use this tool, select the road you want to edit, select a cross section, and adjust the banking by using the 2D Cross section editor window.



Use Slip Connections

RoadRunner offers a way to enforce height constraints between roads that have a dependent relationship, such as a freeway and a freeway offramp. By creating slip roads, the end height and slope of the slip road is constrained to that of the main road. To build a slip road, use the **Slip Road Tool** to pull a slip road off of any other road in your scene.



How Surfaces Work in RoadRunner

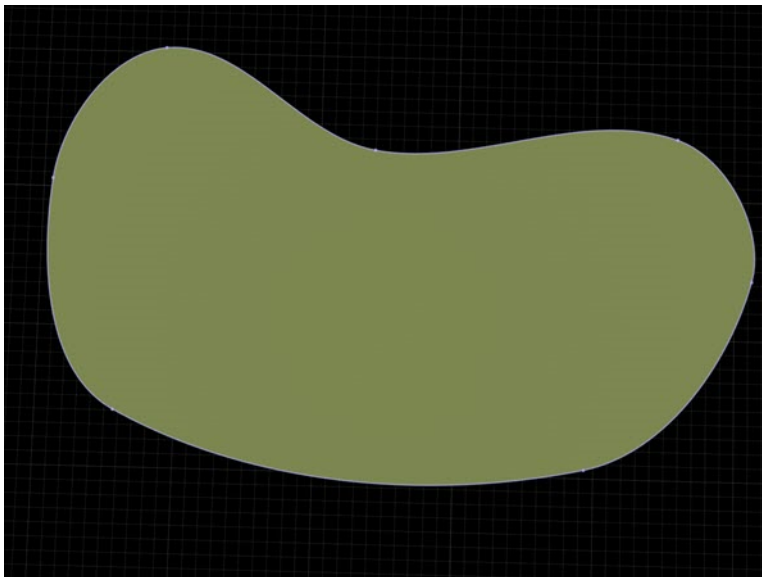
Using the **Surface Tool**, you can model surfaces around roads, such as walkways, driveways, parking lots, and natural terrain. The terrain surface model interacts differently with various aspects of a scene.

Terrain Surface Model

Terrain surfaces are region graphs bounded by curves. For more details about region graphs, see “Region Graph Editing” on page 2-65.

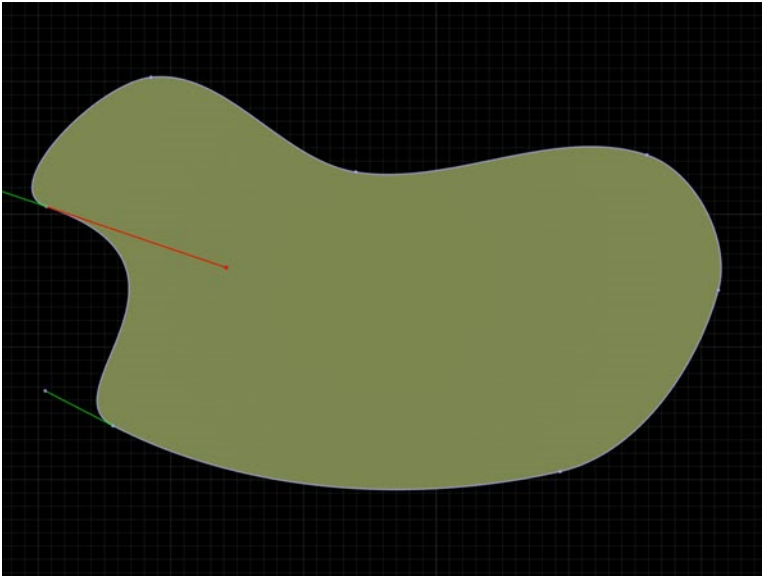
RoadRunner creates some curves automatically, such as the curves on the boundaries of roads. You can create other curves manually by using the **Surface Tool**.

Here is an example of a single terrain surface bounded by manually created surface curves:



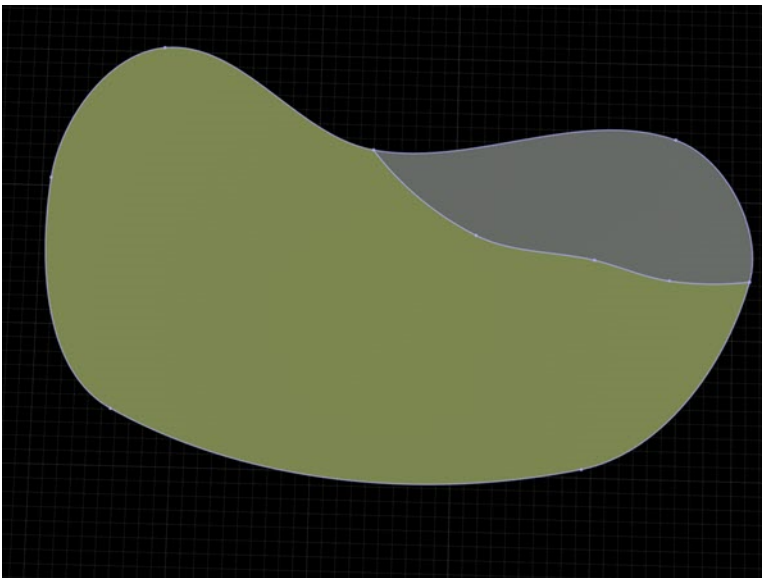
The points on the graph edge curve are curve end nodes, which can be shared by multiple curves. In most regards, these curves use the same UI concepts outlined in the “Curve Editing” on page 2-53 and “Polygon Editing” on page 2-55 topics.

In particular, each curve has a tangent direction that can be modified to change the shape of the curve, as shown in this image:



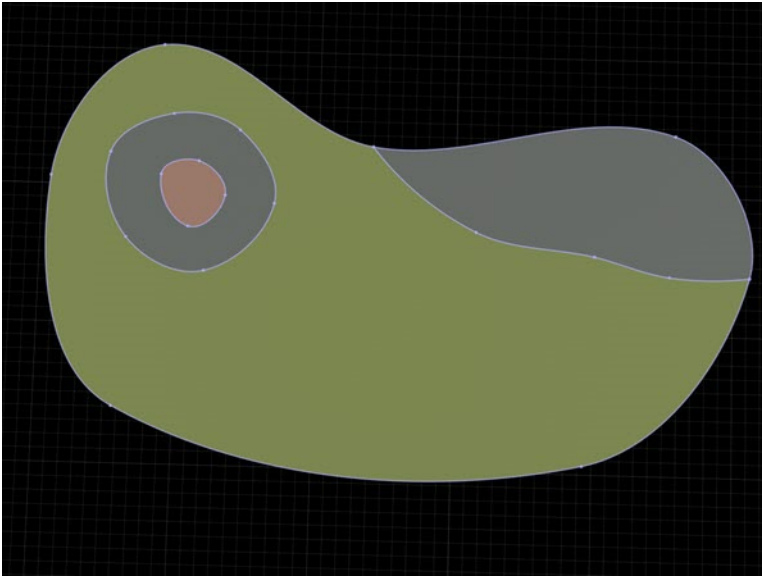
Each surface curve can have one surface connected to each side. The nodes can be shared by any number of surface curves. In this manner, the surface curves form a contiguous (nonoverlapping) patchwork of surfaces called a surface graph.

For example, you can split an initial surface into two surfaces by digitizing new surface curves in the interior, taking care to share end nodes on the perimeter of the surface:



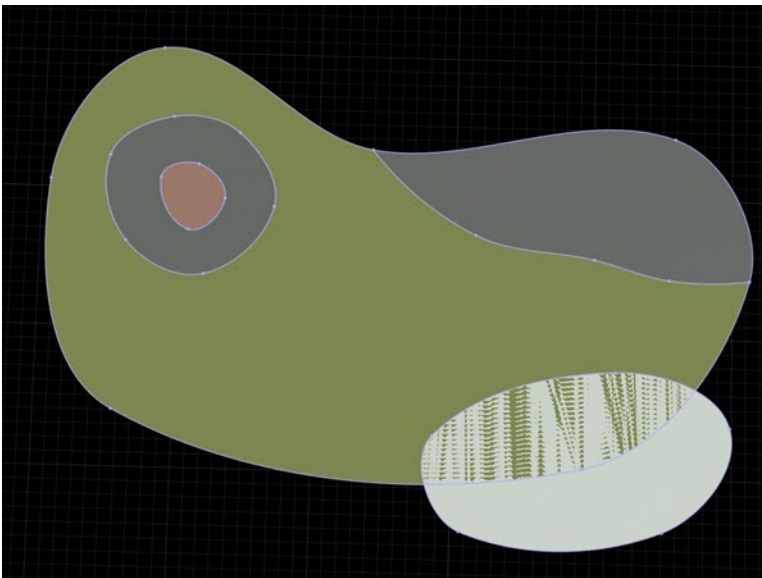
Surfaces also support enclosed surfaces, that is, surfaces within surfaces. Any time a loop of surface curves lies entirely within the interior of another surface, it creates a new surface in the interior.

The following image shows two nested levels of enclosed surfaces:



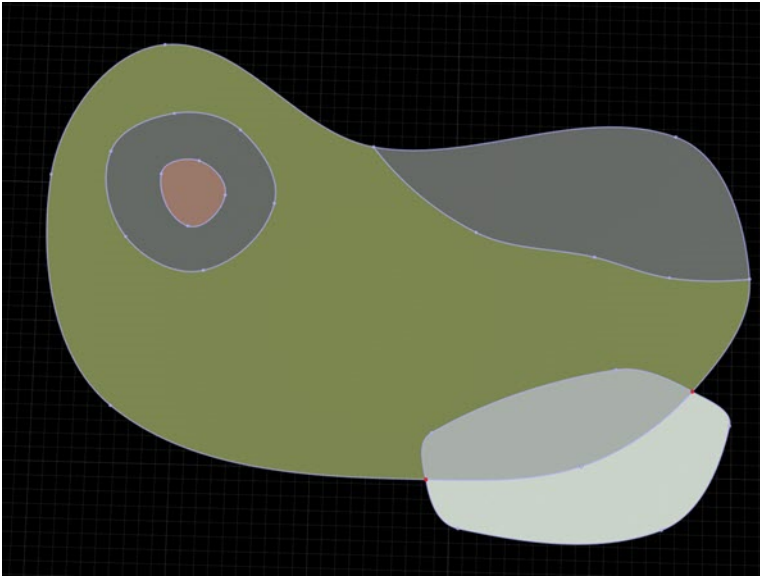
Avoid overlapping surfaces. Surfaces that overlap in the xy dimension cause visual artifacts, or "z-fighting" artifacts.

These artifacts can be seen in the following example, where a new loop of surface curves crosses the existing surface curves:



To correct this issue, you must split the original surface curves to introduce nodes. Multiple surface curves then share these nodes

This image includes corrected nodes.

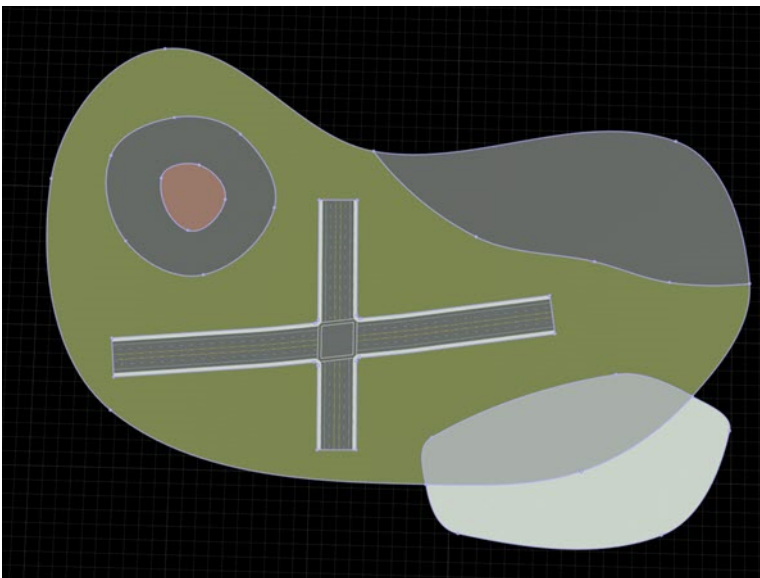


Surfaces and Roads

Roads automatically participate in the surface graph.

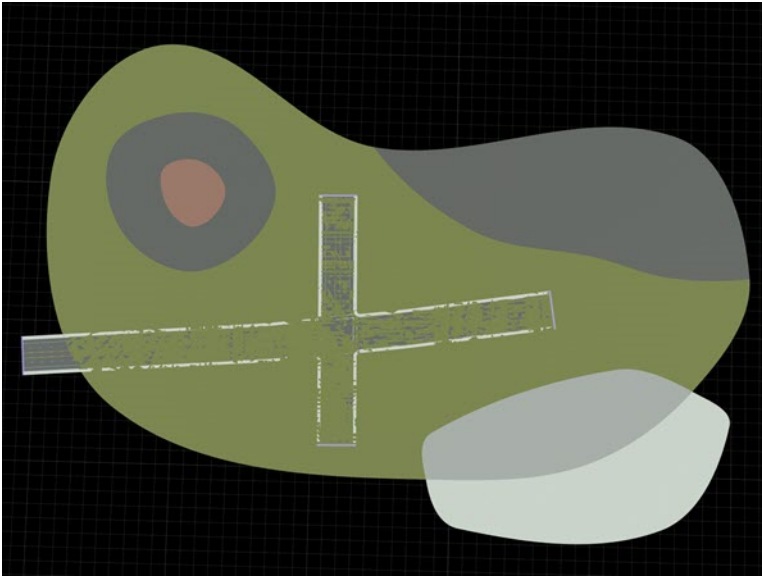
Roads that lie entirely within a terrain surface behave much like enclosed surfaces. Terrain surface curves are automatically created around the perimeter of the road network, forming an enclosed road surface.

In the following image, a simple intersection that was created using the **Road Plan Tool** has been digitized in the interior of the surface:

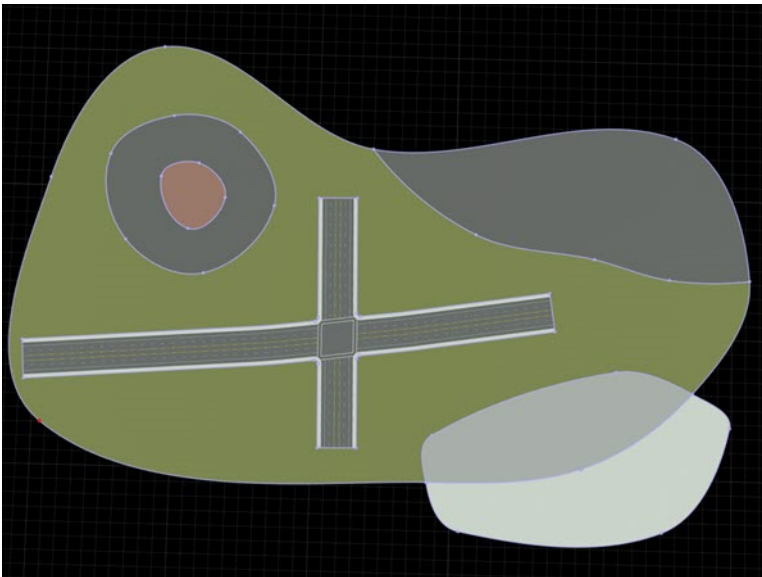


As with overlapping surfaces, roads that overlap a surface curve can cause visual artifacts.

For example, dragging the end of a road such that it crosses a surface curve causes artifacts:

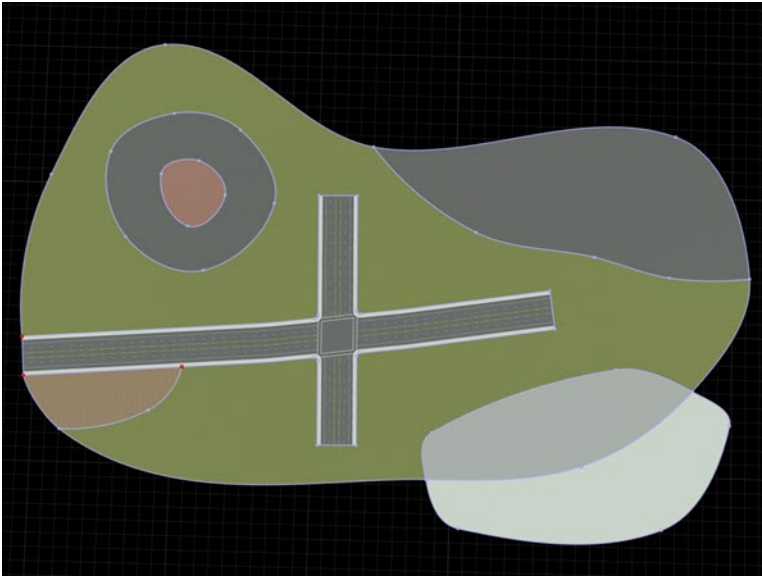


One way to correct this issue is to adjust the containing surface such that the roads are fully enclosed:



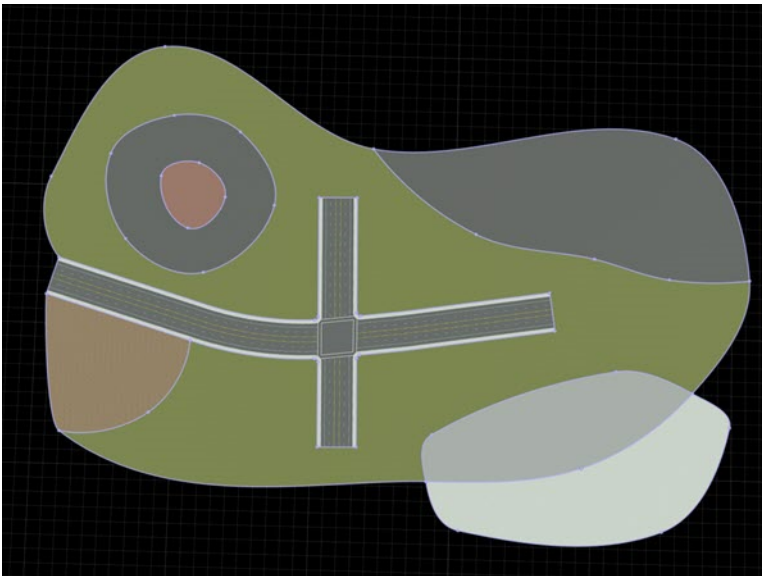
You can also connect surfaces directly to roads. This connection enables surfaces to automatically move when the roads are moved, helping to avoid overlaps.

In the following image, the three nodes attached to the roads are highlighted in red. The two nodes at the end of the road are created automatically and cannot be removed or deleted in the **Surface Tool**. Nodes can also be added parametrically along the side of a road. These points can be inserted anywhere along a road, dragged along the road, or deleted.



When surfaces are attached to these road surface nodes, moving the road automatically adjusts the surfaces.

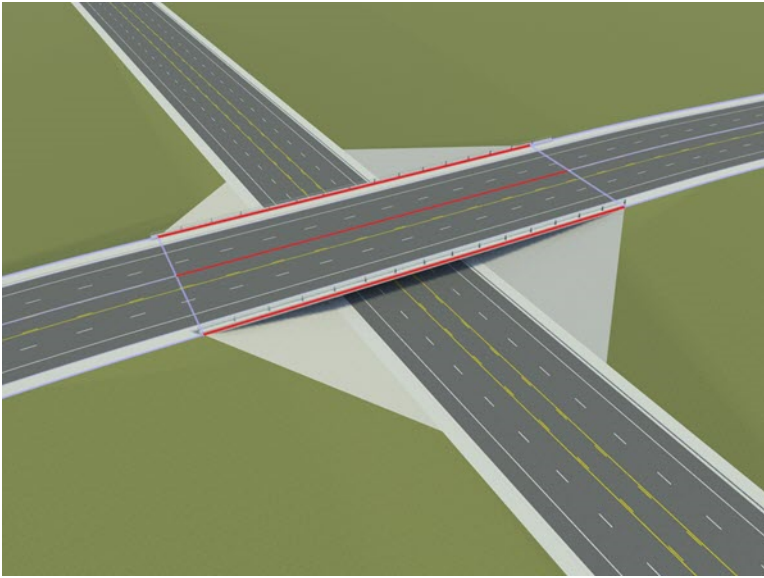
In the following image, the end of the road is moved clockwise:



Bridges

Only nonbridge portions of road surfaces participate in the surface graph. For more details, see the **Road Construction Tool**. The surface graph ignores road construction spans that are marked as bridges.

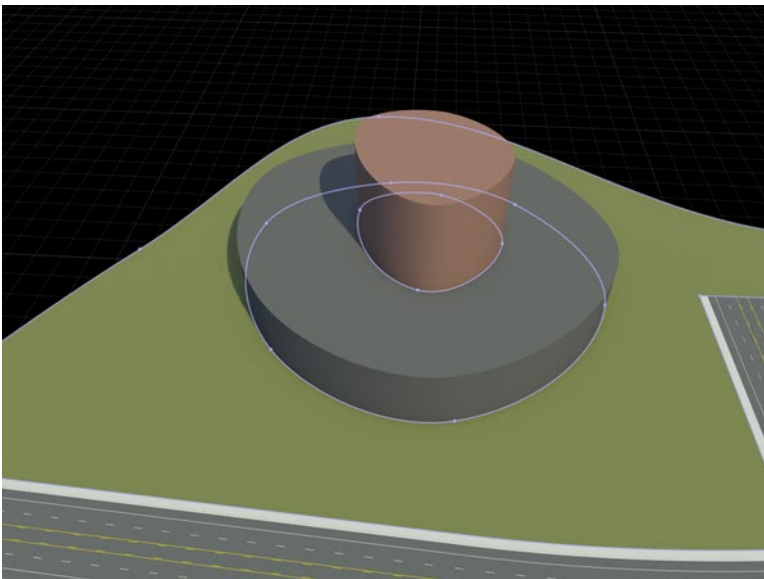
For example, the selected construction span in the following image is marked as a bridge:



Note A bridge span that has similar elevation to the surfaces underneath it can produce visual artifacts. To avoid artifacts, use the **Road Height Tool** to change the elevation of the bridge span so that it is above the ground surface beneath it.

Extruded Surfaces

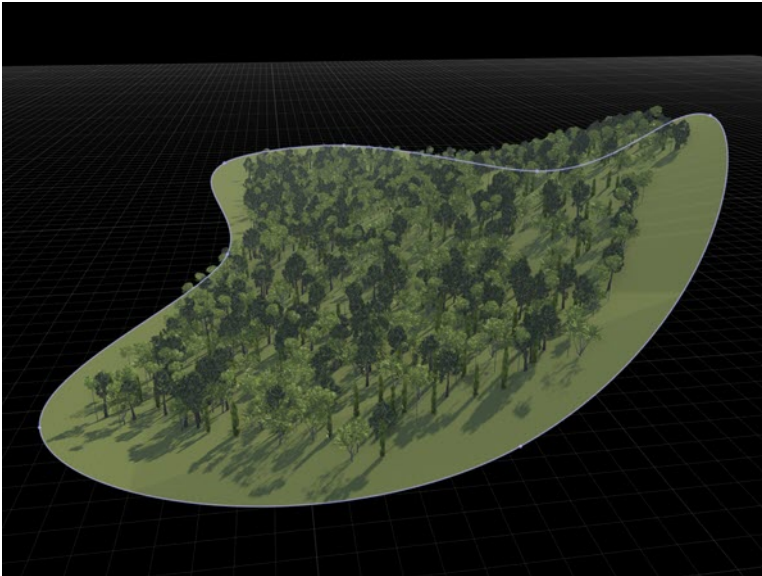
Surfaces have an optional height attribute. If the height is nonzero, the surface is vertically extruded upwards. By increasing the height, you can create simple mock buildings, as shown here:



Surfaces and Elevation

By default, the heights within a surface are automatically interpolated from the heights of the surrounding surface curves.

For example, this image shows a surface containing curves that have a nonzero height, represented as a positive z-coordinate:



In scenes with elevation maps, that is, **Elevation Map Assets**, each surface can optionally use the elevation maps to define the interior elevations of the surface. For more details, see [Control Whether a Surface Uses Elevation Samples](#).

See Also

[Lane Marking Tool](#) | [Prop Point Tool](#) | [Prop Polygon Tool](#) | [Surface Tool](#)

Related Examples

- “Region Graph Editing” on page 2-65

Export Scenes

- “Export to AutoCAD” on page 5-2
- “Export to FBX” on page 5-3
- “Export to glTF” on page 5-5
- “Export to OpenFlight” on page 5-6
- “Export to OpenSceneGraph” on page 5-7
- “Export to Wavefront OBJ” on page 5-8
- “Export to GeoJSON” on page 5-9
- “Export to USD” on page 5-14
- “Export to OpenDRIVE” on page 5-15
- “Left-Hand Drive Export to OpenDRIVE 1.4” on page 5-24
- “Segmentation” on page 5-27
- “Downloading Plugins” on page 5-29
- “RoadRunner Metadata Export” on page 5-30
- “Export to Apollo” on page 5-32
- “Export to Metamoto” on page 5-36
- “Export to Unity” on page 5-37
- “Export to Unreal” on page 5-54
- “Export to CARLA” on page 5-62
- “Export to VTD” on page 5-72
- “Customize Levels of Detail in Exported Scenes” on page 5-77
- “Export Custom Formats” on page 5-88

Export to AutoCAD

You can export RoadRunner scenes to the AutoCAD DXF file format.

AutoCAD Export

From the **File** menu, select **Export**, then **AutoCAD (.dxf)** to open the Export AutoCAD dialog box. Then, specify a path to which to export the file, and click **Export**. Before exporting, you can optionally set these parameters.

Split by Segmentation

Split meshes by their segmentation type. For more details, see Segmentation on page 5-27.

Power of Two Texture Dimensions

Resize the dimensions of exported textures by rounding them up to the next highest power of two.

Embed Textures

Embed the exported textures inside the exported file.

Export to Tiles

Split the meshes per tile. This parameter also groups props by the tile that they are in.

- By default, only one file is exported. Tiles are stored in separate nodes.
- The **Tile Size** parameter specifies the (x,y) dimensions of the exported tiles. Units are in pixels.
- The **Tile Center** parameter specifies the (x,y) location of the tile centers, which is (0,0) by default.
- If you enable the **Export Individual Tiles** parameter, then RoadRunner exports each tile as a separate file. The files follow this naming convention: *ExportedFileName_Tile_0_0.ext*, *ExportedFileName_Tile_1_0.ext*, and so on.

Export to FBX

RoadRunner can export scenes to the FBX file format. Although this option is compatible with Unity and Unreal®, for those applications, using the specific export option along with MathWorks plugins is recommended.

FBX Export

From the **File** menu, select **Export**, then **Filmbox (.fbx)** to open the Export Filmbox dialog box. Then, specify a path to which to export the file, and click **Export**. Before exporting, you can optionally set these parameters.

Split by Segmentation

Split meshes by their segmentation type. For more details, see Segmentation on page 5-27.

Power of Two Texture Dimensions

Resize the dimensions of exported textures by rounding them up to the next highest power of two.

Embed Textures

Embed the exported textures inside the exported file.

Export to Tiles

Split the meshes per tile. This parameter also groups props by the tile that they are in.

- By default, only one file is exported. Tiles are stored in separate nodes.
- The **Tile Size** parameter specifies the (x,y) dimensions of the exported tiles. Units are in pixels.
- The **Tile Center** parameter specifies the (x,y) location of the tile centers, which is (0,0) by default.
- If you enable the **Export Individual Tiles** parameter, then RoadRunner exports each tile as a separate file. The files follow this naming convention: *ExportedFileName_Tile_0_0.ext*, *ExportedFileName_Tile_1_0.ext*, and so on.

Advanced Details

Node Naming

- All nodes have "Node" appended to the end of their name (for example, *RoadsNode*).
- Props have their GUID prepended to the front of their name (for example, *{40ce66ca-c817-425b-8802-17cdbc76371f}Signal_Post_30ftNode*).
 - Props generated from a curve, polygon, or span share GUIDs with the associated curve, polygon, or span.
- During export, all node names in the scene graph are made to be unique.
 - *_#* is appended for duplicate node names (for example, *light_green_1Node*).
 - Because the scene graph is not a tree, duplicate names are still possible when converting to FBX if there are multiple instances of the same node (such as when reusing props).

- When the mesh is split by segmentation, the roads and terrain have extra child nodes for each segmentation type it has, with the segmentation type appended to the name (for example, `Road_SidewalkNode`).
- When the mesh is split by transparency sorting layer (for the Unreal on page 5-54 export option), the roads and terrain have extra child nodes for each sorting layer, with the layer number appended to the name (for example, `Roads_Layer2Node`).
 - This can also be combined with the segmentation type (for example, `Road_Sidewalk_Layer0Node`).
- For traffic signals (see **Signal Tool**), the name of the variant is added to the FBX name (for example, `{4674ef2e-deea-403c-9b52-487e0ba9f13d}Signal_3Light_Bare01_RedYellowGreen_LeftNode`).

Material Details

- Materials are converted into `FbxSurfacePhong` materials.
- When the mesh is split by segmentation, the segmentation type is appended to the material name (for example, `Concrete1_Curb`).
- When materials need a transparency sorting order defined (typically for overlapping transparent markings), a duplicate of the material is created.
- Materials with duplicate names add `_#` to distinguish between them (for example, `Leaves_1`).
 - This can be combined with the segmentation type (for example, `OilPath01_Road_1`).

Light Source Parameters

When you import FBX file assets containing light sources, the parameters that control lighting effects, such as color and brightness, are carried through to exported FBX files.

FBX Scene Settings

RoadRunner exports scenes with the Maya Z-up axis system. Units are in meters.

Combo Exports

Other export options combine the FBX export with other files. Depending on the target application, the RoadRunner software applies extra changes.

- Unity on page 5-37: Mesh is identical to the normal FBX export option.
- Unreal on page 5-54: Mesh is split by transparency sorting order.
- CARLA on page 5-62: Mesh is split by segmentation and transparency sorting order.

See Also

Related Examples

- “Customize Levels of Detail in Exported Scenes” on page 5-77

Export to glTF

RoadRunner can export scenes to the GL Transmission Format (glTF).

glTF Export

From the **File** menu, select **Export**, then **glTF (.gltf)** to open the Export glTF dialog box. Then, specify a path to which to export the file, and click **Export**. Before exporting, you can optionally set these parameters.

Split by Segmentation

Split meshes by their segmentation type. For more details, see Segmentation on page 5-27.

Power of Two Texture Dimensions

Resize the dimensions of exported textures by rounding them up to the next highest power of two.

Export to Tiles

Split the meshes per tile. This parameter also groups props by the tile that they are in.

- By default, only one file is exported. Tiles are stored in separate nodes.
- The **Tile Size** parameter specifies the (x,y) dimensions of the exported tiles. Units are in pixels.
- The **Tile Center** parameter specifies the (x,y) location of the tile centers, which is (0,0) by default.
- If you enable the **Export Individual Tiles** parameter, then RoadRunner exports each tile as a separate file. The files follow this naming convention: *ExportedFileName_Tile_0_0.ext*, *ExportedFileName_Tile_1_0.ext*, and so on.

Limitations

RoadRunner follows the glTF 2.0 specification as much as possible, but there are some limitations.

- Texture sampler information is not exported. This limitation might result in texture clamping issues for objects like signs.
- RoadRunner uses a specular setup for materials. Because glTF uses metallic-roughness by default, RoadRunner attaches whatever is in the "Specular Map" slot to the "metallicRoughnessTexture" in the exported material and sets the "metallicFactor" to 0.

Export to OpenFlight

RoadRunner can export scenes to the OpenFlight (.flt) file format.

OpenFlight Export

From the **File** menu, select **Export**, then **OpenFlight (.flt)** to open the Export OpenFlight dialog box. Then, specify a path to which to export the file, and click **Export**. Before exporting, you can optionally set these parameters.

Split by Segmentation

Split meshes by their segmentation type. For more details, see Segmentation on page 5-27.

Power of Two Texture Dimensions

Resize the dimensions of exported textures by rounding them up to the next highest power of two.

Export to Tiles

Split the meshes per tile. This parameter also groups props by the tile that they are in.

- By default, only one file is exported. Tiles are stored in separate nodes.
- The **Tile Size** parameter specifies the (x,y) dimensions of the exported tiles. Units are in pixels.
- The **Tile Center** parameter specifies the (x,y) location of the tile centers, which is (0,0) by default.
- If you enable the **Export Individual Tiles** parameter, then RoadRunner exports each tile as a separate file. The files follow this naming convention: *ExportedFileName_Tile_0_0.ext*, *ExportedFileName_Tile_1_0.ext*, and so on.

Limitations

- Texture wrapping settings are not exported. This limitation might result in texture clamping issues for objects like signs.
- The OpenFlight file is exported through the OpenSceneGraph plugin, which exports OpenFlight version 16.1.
- RoadRunner does not export the normal or specular maps for materials.

Export to OpenSceneGraph

RoadRunner can export scenes to various OpenSceneGraph file formats, including `.osg`, `.osgb`, and `.ive`.

OpenSceneGraph Export

From the **File** menu, select **Export**, then **OpenSceneGraph (.osg, .osgb, .ive)** to open the Export OpenSceneGraph dialog box. Then, specify a path to which to export the file, and click **Export**. Before exporting, you can optionally set these parameters.

Split by Segmentation

Split meshes by their segmentation type. For more details, see Segmentation on page 5-27.

Power of Two Texture Dimensions

Resize the dimensions of exported textures by rounding them up to the next highest power of two.

Embed Textures

Embed the exported textures inside the exported file.

Export to Tiles

Split the meshes per tile. This parameter also groups props by the tile that they are in.

- By default, only one file is exported. Tiles are stored in separate nodes.
- The **Tile Size** parameter specifies the (x,y) dimensions of the exported tiles. Units are in pixels.
- The **Tile Center** parameter specifies the (x,y) location of the tile centers, which is (0,0) by default.
- If you enable the **Export Individual Tiles** parameter, then RoadRunner exports each tile as a separate file. The files follow this naming convention: *ExportedFileName_Tile_0_0.ext*, *ExportedFileName_Tile_1_0.ext*, and so on.

Limitations

- Texture wrapping settings are not exported. This limitation might result in texture clamping issues for objects like signs.
- RoadRunner does not export the normal or specular maps for materials.

Export to Wavefront OBJ

RoadRunner can export scenes to the Wavefront OBJ (.obj) file format.

Wavefront Export

From the **File** menu, select **Export**, then **Wavefront (.obj)** to open the Export Wavefront dialog box. Then, specify a path to which to export the file, and click **Export**.

Advanced Details

This code shows an example of a material in an exported Material Library (.mtl) file:

```
newmtl Grass1           # Material Name
illum 2                 # Color with specular highlights
Ka 1.000000 1.000000 1.000000 # Ambient color matches the diffuse
Kd 1.000000 1.000000 1.000000 # Diffuse color
Ks 0.039216 0.039216 0.039216 # Specular color
Ns 800.200012          # Specular exponent, approximated from the Roughness value
Tr 0.000000            # Transparency
map_Kd Grass1_Diff.png # Diffuse Map
map_bump Grass1_Norm.png # Normal Map
map_Ks Grass1_Spec.png # Specular Map
```

Export to GeoJSON

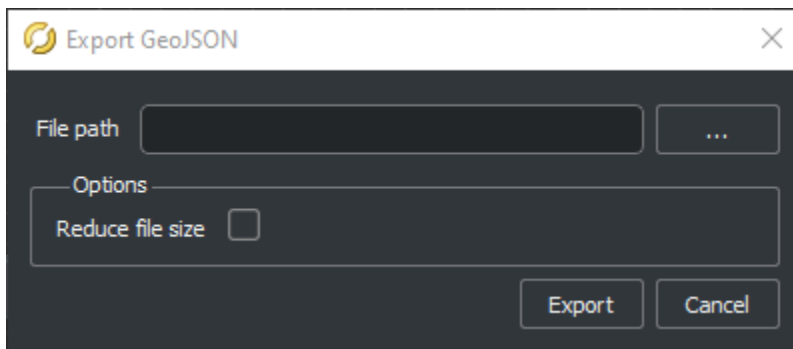
RoadRunner can export scenes to a GeoJSON file format. This format is meant to complement the OpenDRIVE file format and fill in some of its missing data, but it can also be used on its own.

GeoJSON Export

From the menu, select **File > Export > GeoJSON (.geojson)**.

Export Options

Set the file destination and click **Export**.



Select **Reduce file size** to remove new lines from the exported GeoJSON file and decrease its size.

Sample GeoJSON File

The MathWorks version of the GeoJSON format is a collection of lanes, lane boundaries, junctions, gates, crosswalks, and signals.

This code shows a sample GeoJSON file.

```
'#' are double values compliant with JSON
{
  "features": [{
    "geometry": {
      "coordinates": [[#, #, #], ...],
      "type": "LineString"
    },
    "properties": {
      "Id": 1,
      "LaneType": "Curb",
      "LeftBoundary": {
        "Dir": "Forward",
        "Id": 0
      },
      "Predecessors": [{
        "Dir": "Forward",
        "Id": 4
      } ],
      "RightBoundary": {
        "Dir": "Forward",
        "Id": 5
      },
      "Successors": [{
        "Dir": "Backward",
```

```

        "Id": 6
      },
      "TravelDir": "Undirected",
      "Type": "Lane"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [[#, #, #], ...],
      "type": "LineString"
    },
    "properties": {
      "Id": 0,
      "LeftLane": {
        "Dir": "Forward",
        "Id": 1
      },
      "RightLane": {
        "Dir": "Forward",
        "Id": 2
      },
      "Type": "LaneBoundary"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [[[#, #, #], ...], ...], ...],
      "type": "MultiPolygon"
    },
    "properties": {
      "Id": 12,
      "Type": "Junction",
      "Gates": [{"Id": 775}, ...],
      "Lanes": [{"Id": 52}, ...],
      "Phases": [
        "Phases": [{
          "Intervals": [{
            "BulbStates": [{"Id": 0, "On": false, "SignalId": 767}, ...],
            "GateStates": [{"Id": 775, "State": "StopYield"}, ...]
          }
        ]
      ]
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [[#, #, #], ...],
      "type": "LineString"
    },
    "properties": {
      "Id": 775,
      "Lane": {"Id": 233},
      "Signals": [{"Id": 771}, ...],
      "Type": "Gate"
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [[[#, #, #], ...], ...],
      "type": "Polygon"
    },
    "properties": {
      "Id": 763,
      "Bulbs": [
        {
          "Point": [#, #, #],
          "Name": "LeftTurnRed",
          "NodeName": "light_red"
        }
      ]
    }
  }
}

```

```

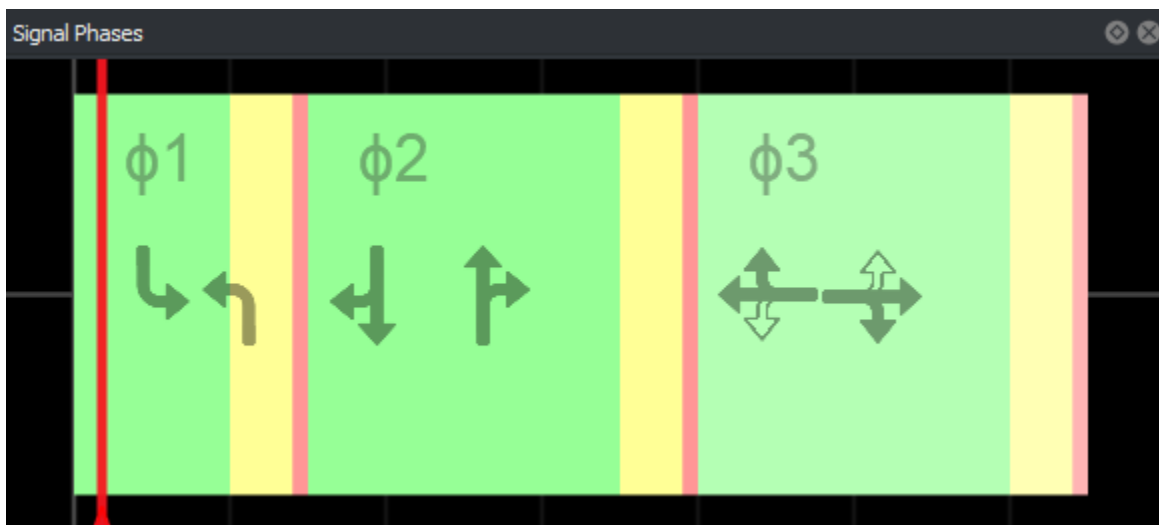
    },
    ""
  ],
  "Name": "PathToFile.fbx",
  "SignalType": "Signal",
  "Type": "Signal"
},
"type": "Feature"
}]
}

```

Traffic Signal Phases in GeoJSON

Most of the information in the GeoJSON files describe the geometry of a scene. The Phases sections of these files describe the traffic light phases of signals at each junction.

Consider a junction with three signal phases, with each one containing green-yellow-red intervals of varying durations. This figure shows such a sample phase as it appears in the **2D Editor**.



This GeoJSON snippet corresponds to the first traffic phase. Some of the data for this phase has been omitted for clarity.

```

"Phases": [
  {
    "Intervals": [
      {
        "BulbStates": [
          {
            "Id": "{8bedd7ab-7e17-4177-b095-ddca457e6985}",
            "On": false,
            "SignalId": "{cc65a9c4-f47c-465b-8a8f-ae7c7e2aca50}"
          },
          .
          .
          .
        ],
        "GateStates": [
          {
            "Id": "{3233fba0-a5b6-4f09-a442-2e0ddc07c4c0}",
            "State": "Go"
          }
        ]
      }
    ]
  }
]

```

```

    },
    :
    :
  ],
  "Time": 10,
  "Type": "Green"
},
{
  "BulbStates": [
    {
      "Id": "{8bedd7ab-7e17-4177-b095-ddca457e6985}",
      "On": false,
      "SignalId": "{cc65a9c4-f47c-465b-8a8f-ae7c7e2aca50}"
    },
    :
    :
  ],
  "GateStates": [
    {
      "Id": "{3233fba0-a5b6-4f09-a442-2e0ddc07c4c0}",
      "State": "Go"
    },
    :
    :
  ],
  "Time": 4,
  "Type": "Yellow"
},
{
  "BulbStates": [
    {
      "Id": "{8bedd7ab-7e17-4177-b095-ddca457e6985}",
      "On": true,
      "SignalId": "{cc65a9c4-f47c-465b-8a8f-ae7c7e2aca50}"
    },
    :
    :
  ],
  "GateStates": [
    {
      "Id": "{3233fba0-a5b6-4f09-a442-2e0ddc07c4c0}",
      "State": "Stop"
    },
    :
    :
  ],
  "Time": 1,
  "Type": "Red"
}
]
},
{
  "Intervals": [

```



```
    .  
    .  
    .  
  ]
```

The `Time` and `Type` values specify that the first phase has a green interval that lasts 10 seconds, a yellow interval that lasts 4 seconds, and a red interval that lasts 1 second.

Each `BulbStates` section lists the unique ID of a traffic light bulb in the scene. In full GeoJSON files, you can find the bulb specification by searching for this ID. In the first (green) interval, the first bulb in `BulbStates` has an ID of `8bedd7ab-7e17-4177-b095-ddca457e6985` with its `"On"` state set to `false`. In the full GeoJSON file (not shown here), this bulb is specified as being light red. Therefore, it is expected that the bulb is off at this phase. In the GeoJSON file snippet, you can see that the bulb with this ID is also off in the second (yellow) interval but is then on in the third (red) interval.

Each `GateStates` section lists the unique ID of a maneuver gate in the junction and what its state is during each interval. As with the bulbs in the `BulbStates` sections, the gates in the `GateStates` sections are repeated in each interval, with only the states differing at each interval.

Though not shown in this GeoJSON snippet, the full GeoJSON file includes two additional `Intervals` sections for the two remaining intervals in the phase.

Export to USD

RoadRunner can export to the Universal Scene Description (USD) file format.

USD Export

From the **File** menu, select **Export**, then **USD (.usd, .usdc, .usda)** to open the Export USD dialog box.

Specify a path to which to export the file, and click **Export**. Setting the file extension to `.usd` or `.usdc` creates a USD binary file. Setting the extension to `.usda` creates a USD ASCII file.

Before exporting, you can optionally set these parameters.

Split by Segmentation

Split meshes by their segmentation type. For more details, see Segmentation on page 5-27.

Power of Two Texture Dimensions

Resize the dimensions of exported textures by rounding them up to the next highest power of two.

Export to Tiles

Split the meshes per tile. This parameter also groups props by the tile that they are in.

- By default, only one file is exported. Tiles are stored in separate nodes.
- The **Tile Size** parameter specifies the (x,y) dimensions of the exported tiles. Units are in pixels.
- The **Tile Center** parameter specifies the (x,y) location of the tile centers, which is (0,0) by default.
- If you enable the **Export Individual Tiles** parameter, then RoadRunner exports each tile as a separate file. The files follow this naming convention: *ExportedFileName_Tile_0_0.ext*, *ExportedFileName_Tile_1_0.ext*, and so on.

Limitations

There are several limitations with the `UsdPreviewSurface` schema, which limits the ability to map RoadRunner materials to USD materials. Here is a list of known issues:

- `UsdPreviewSurface` cannot support diffuse color scaling and vertex coloring at the same time. RoadRunner sets the texture reader's scale to the diffuse color of the material. However, vertex colors and a `primvar` reader for them are still included if you want to use them.
- Because `UsdPreviewSurface` does not support specifying a rendering order, overlapping transparent surfaces are not supported.
- Because the double-sided attribute is stored per mesh instead of per material, it is not exported. Therefore, some props (mainly trees) might not render properly.

Export to OpenDRIVE

RoadRunner can export scenes to the OpenDRIVE (.xodr) file format.

OpenDRIVE Overview

RoadRunner can export scenes to the OpenDRIVE 1.4 format. The OpenDRIVE export option exports an OpenDRIVE (.xodr) file containing the layout of the scene and an optional MathWorks GeoJSON file.

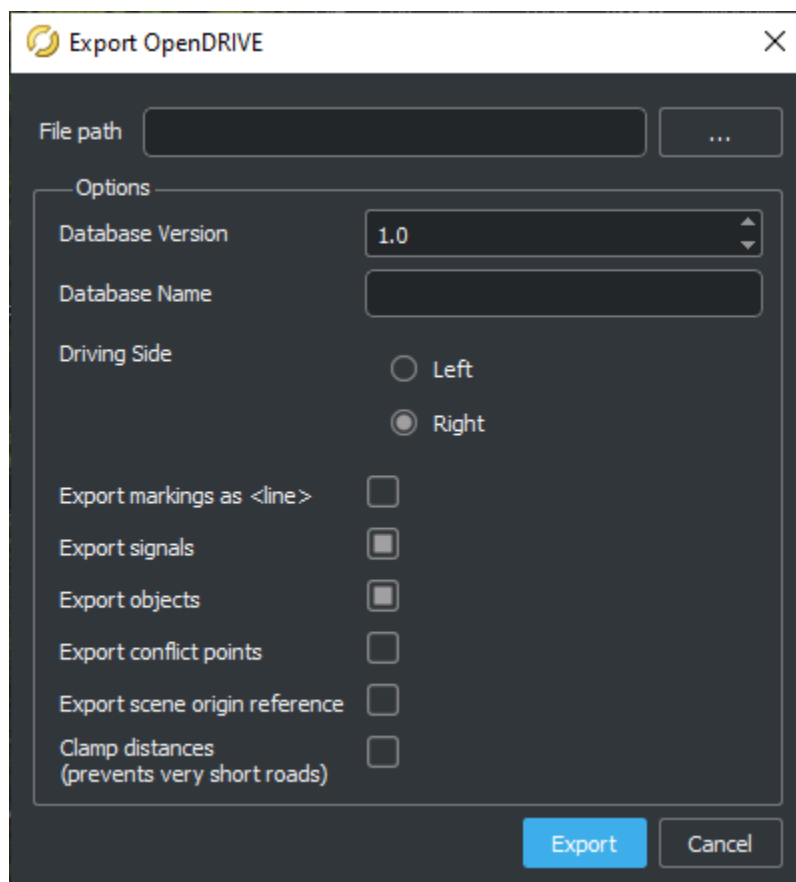
Export to OpenDRIVE

From the menu, select **File > Export > OpenDRIVE (.xodr)**.

To preview the OpenDRIVE export and explore exported data interactively, use the **OpenDRIVE Export Preview Tool**

Export Options

The Export OpenDRIVE dialog box has several options to conform to various simulator needs.



| Attribute | Description |
|--|---|
| Database Version | A user-defined identifier for the exported scene. Useful for versioning exports of the same scene. |
| Database Name (optional) | A user-defined name for the exported scene. |
| Driving Side | <p>A hint to the exporter for the driving side of the scene. Travel direction is explicitly defined in RoadRunner using the Lane Travel Direction.</p> <p>The travel direction in OpenDRIVE is implicit based on the country and side of the road.</p> |
| Export markings as <line> | Select this attribute to export additional lane marking data (spacing, dash length, and individual paint strip widths). |
| Export signals | <p>Select this attribute to export all signals and signs mapped to junctions as <signal> entries.</p> <p>This selection applies only to signals and signs that have been associated with junctions (using the Signal Tool). Refer to the Traffic Signals and Signs on page 5-20 section.</p> |
| Export objects | Select this attribute to export all props as <object> entries. Refer to the “OpenDRIVE User Asset Configuration” on page 5-22 section. |
| Export conflict points | Select this attribute to export an <object> entry for every point in a junction where two roads intersect. |
| Export scene origin reference | Select this attribute to export a point at 0,0 in the scene. This point enables a connection between FLT or IVE files and the exported OpenDRIVE file. This point is contained in a <road> entry with no lanes that is positioned at the far left edge of the scene, nested within an <object> entry. |
| Clamp distances (prevents very short roads) | <p>Select this attribute to clamp distances in the RoadRunner scene to be a multiple of 1 cm to prevent very short roads.</p> <hr/> <p>Note This clamping is performed on the scene itself, so it can cause very small changes to the roads in the scene.</p> |

OpenDRIVE Representations

This section describes how various types of RoadRunner objects are represented in OpenDRIVE.

Roads, Lanes, and Junctions



Roads, lanes, and junctions are exported to OpenDRIVE using the standard `<road>`, `<lane>`, and `<junction>` entries.

For each road in a scene, RoadRunner creates one or more `<road>` entries. Whenever a road ends or a junction begins or ends, RoadRunner creates a unique `<road>` entry. OpenDRIVE `<road>` entries cannot extend through a junction, so the geometry is cut and exported as separate roads.

Note The `<shape>`, `<crossfall>`, `<surface>`, and `<railroad>` entries are not used.

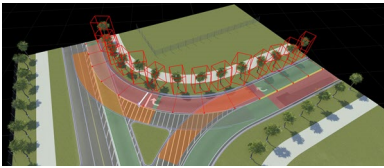
For each lane in a scene, RoadRunner creates one or more `<lane>` entries. The resulting `<lane>` entry is placed on one side or the other of the center lane, depending on its travel direction or the travel direction of neighboring lanes and the selected **Driving Side** during export. Whenever a lane starts or ends, RoadRunner creates a new `<laneSection>` entry.

Note The `level` flag in `<lane>` entries is not used. The `<height>`, `<material>`, `<visibility>`, and `<access>` entries are also not used.

For each junction in a scene, RoadRunner creates a `<junction>` entry. RoadRunner exports some junctions as one `<junction>` entry due to overlapping maneuver roads or corners. A connecting `<road>` entry is exported for each maneuver road in each junction. Where possible, the exporter prefers the geometry and lane markings of nonmaneuver roads that extend through the junction. The resulting geometry of each connecting road might be the combination of multiple maneuver and nonmaneuver roads.

Note The `<priority>` entry is not used.

Props

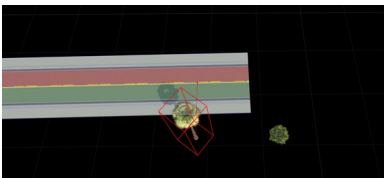


Note Prop polygons are not exported, but if you run the **bake** operation to convert them to points, you can export them in point format. See **Prop Polygon Tool**.

With the exception of traffic signals and signs (see below), point props are exported as OpenDRIVE <object> instances. The exported prop includes sufficient information to identify the prop type and the oriented bounding box (OBB) of the prop model.

In OpenDRIVE, objects are stored on roads. The position and orientation of a given object depends on the geometry of the road it is assigned to. RoadRunner props are freely positioned in the world, so the export process must choose a road for each prop to export. In most cases, RoadRunner selects the road closest to the prop.

Note In some cases, it is impossible to represent a prop's position in OpenDRIVE. In this image, the bush on the right is past the end of the road and there is no other road in the scene. In this case, the prop is not exported and a warning is displayed during export.

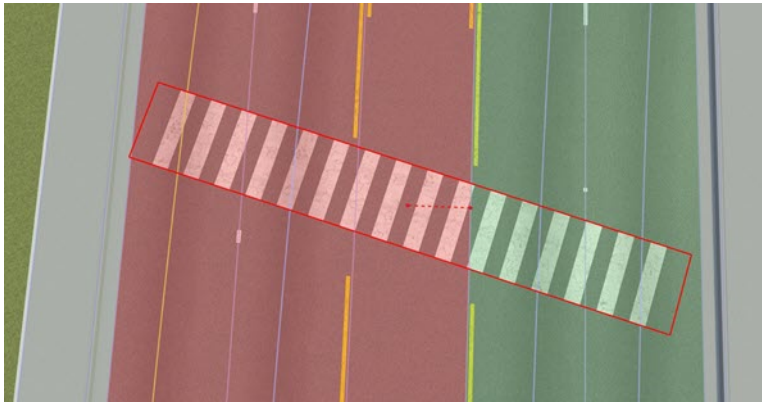


Prop Attributes

Exported props include the following attributes:

| OpenDRIVE Attribute | Description |
|---------------------|--|
| name | Name of the prop asset (for example, "Signal_Post_30ft") |
| s/t | Inertial position of the prop point |
| hdg/roll/pitch | Inertial rotations of the prop point |
| zOffset | Relative height of the prop point |
| height/width/length | Dimensions of the prop model's bounding box |
| type | Object type, as defined by the configuration XML file for the point's asset (refer to "OpenDRIVE User Asset Configuration" on page 5-22) |

Crosswalks and Marking Polygons



Crosswalks and marking polygons are exported as OpenDRIVE `<outline>` objects, similar to the crosswalk example in section 7.4 of the OpenDRIVE 1.5M specification.

Unlike that example, RoadRunner exports the polygon vertices as `<cornerLocal>` objects (rather than `<cornerRoad>` objects), which means that the vertices are defined relative to the pivot point specified in the attributes of the `<object>` parent.

This example code shows the representation of the crosswalk polygon in the previous image.

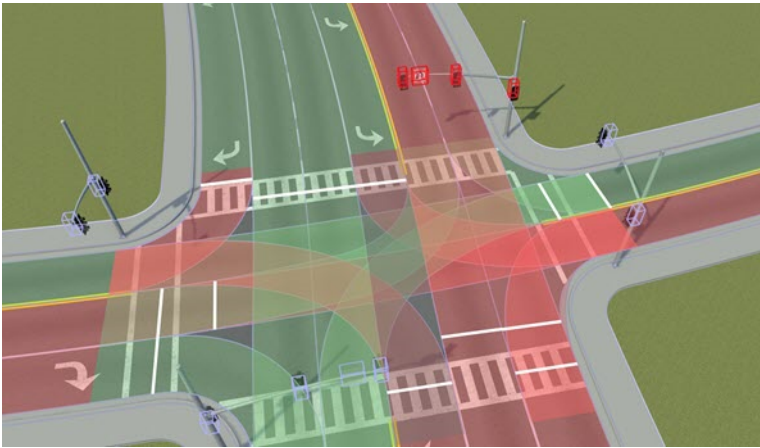
```
<object id="162" name="ContinentalCrosswalk" s="5.9095723267801631e+1" t="1.7834630409170869e+0" zoffset="1.9073486328125000e-6">
  <outline>
    <cornerLocal u="-8.0157129245630365e+0" v="1.4628157932607735e+0" z="0.0000000000000000e+0"/>
    <cornerLocal u="8.0819274304890261e+0" v="1.0363072624453764e+0" z="-1.9073486328125000e-6"/>
    <cornerLocal u="8.0157129245556504e+0" v="-1.4628157932298933e+0" z="-1.9073486328125000e-6"/>
    <cornerLocal u="-8.0819274304964104e+0" v="-1.0363072624144962e+0" z="0.0000000000000000e+0"/>
    <cornerLocal u="-8.0157129245630365e+0" v="1.4628157932607735e+0" z="0.0000000000000000e+0"/>
  </outline>
</object>
```

Crosswalks and Marking Polygon Attributes

Exported crosswalks and marking polygons include the following attributes.

| OpenDRIVE Attribute | Description |
|---------------------|---|
| name | Name of the marking (for example, "ContinentalCrosswalk") |
| s/t | Inertial position of the pivot point |
| hdg/roll/pitch | Inertial rotations of the pivot point |
| zOffset | Relative height of the pivot point |
| height/width/length | Dimensions of an oriented bounding box fit to the polygon's vertices. The 'width' is treated as the dimension along the road, and the 'length' is treated as the dimension across the road. |
| type | Object type, as defined by the configuration XML file for the marking's asset (refer to "OpenDRIVE User Asset Configuration" on page 5-22) |

Traffic Signals and Signs



RoadRunner exports traffic signals and signs as OpenDRIVE `<signal>` objects.

For optimal behavior, traffic signals and signs for controlled intersections should be mapped to junction gates by using the **Signal Tool**. Traffic signals are exported only if they are mapped to junction gates. Signs are exported regardless if they are mapped to junction gates and are automatically mapped to the nearest road if not explicitly mapped.

If you need to add a traffic signal outside of a controlled intersection (for example, for a freeway onramp or pedestrian crossing), you can use the **Custom Junction Tool** to create a junction along a single road.

Note Traffic signals and signs within **Prop Assembly Assets** are not exported. To export signals or signs in an assembly, you must first expand the instance of the assembly.

Signals and Signal References

When a signal or sign is mapped to a junction gate, it appears in the OpenDRIVE export as a `<signal>` instance and one or more `<signalReference>` instances, where:

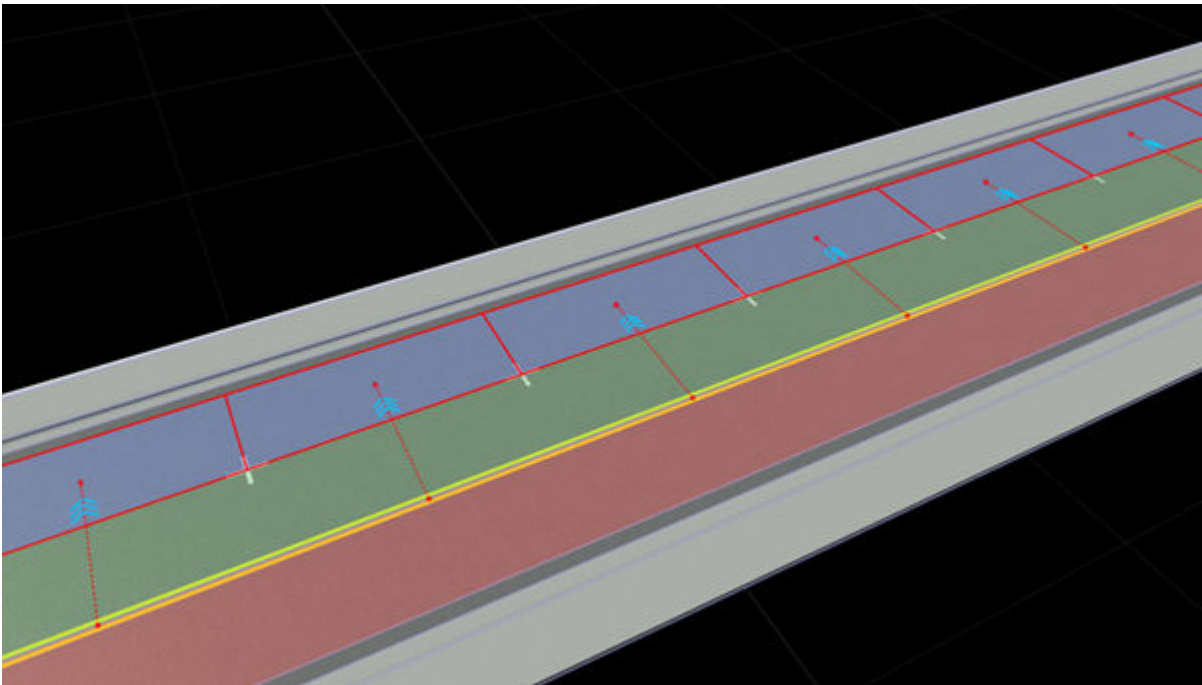
- `<signal>` defines the physical location of the signal. Use `<signal>` to derive the 3D location of the signal, regardless of which roads or lanes are controlled by the signal. In most cases, the signal is mapped to the closest road (similar to the approach used for “Props” on page 5-18). This mapping might have no logical association to the signal (for example, the signal could be a nearby side street).
- `<signalReference>` associates the signal to the roads and lanes that are controlled by the signal. Signal references indicate the semantic relationship between the signal and the road network (as opposed to `<signal>`, which is used purely for geometric positioning). Signal references are present for each maneuver road gate associated with the signal (through the **Signal Tool**).

Signal Attributes

Exported signals and signs include the following attributes:

| OpenDRIVE Attribute | Description |
|---------------------|---|
| name | Refer to Prop Attributes on page 5-18. |
| s/t | "hoffset" is treated the same as the <object> "hdg" attribute. |
| hoffset/roll/pitch | |
| zoffset | |
| height/width/length | |
| type/subtype | Signal type and subtype, as defined by the configuration XML file for the signal's asset (refer to "OpenDRIVE User Asset Configuration" on page 5-22) |
| country | "OpenDRIVE" is always used. |
| dynamic | Specify "yes" for dynamic junction signalization and "no" for static signalization (for example, "All Go" or "All Stop") |
| value | Unused (set to "- 1" in all cases). |
| text | Unused (set to empty string in all cases). |

Parking Spaces



RoadRunner exports parking spaces as an <object> with type "parking" and an additional <parkingSpace> entry under the <object> following section 5.3.8.1.5 in the OpenDRIVE 1.4H specification. Markings on a parking space are exported as <marking> under the <parkingSpace> entry following section 5.3.8.1.6 in the OpenDRIVE 1.4H specification.

Parking Attributes

Exported parking spaces include the following attributes.

| OpenDRIVE Attribute | Description |
|--|--|
| name s/t hdg/roll/pitch zOffset height/width/length | Refer to Prop Attributes on page 5-18. |
| type | Always set to "parking". |
| side (attribute for <marking> entries under <parkingSpace>) | Side of the marking ("left", "right", "front", or "rear"), where the rear is the entry point of the parking space. |
| type/width/color (attributes for <marking> entries under <parkingSpace>) | Same properties as <roadMark> entries for <lane>. |

OpenDRIVE User Asset Configuration

The OpenDRIVE exporter uses a configuration XML file to map RoadRunner props, signals, signs, and markings to the appropriate <object> or <signal> id and subtype. This configuration file is also used to define the correlation during import.

Export a Prop or Signal Type

- 1 Copy the `OpenDriveAssetData.xml` file located in the RoadRunner install location under `AssetsInstall/ResourceAssets` to the Project folder in your project, next to the `Project.rproj` file. For more details on the project folder setup, see "RoadRunner Project and Scene System" on page 2-2.
- 2 Open the new `OpenDriveAssetData.xml` file in a text editor.
- 3 Add entries for new objects, markings, or signals.
- 4 Save your file and export an OpenDRIVE file.

You do not need to restart RoadRunner after creating or modifying the `OpenDriveAssetData.xml` file.

Here is the definition of the format of the `OpenDriveAssetData.xml` file:

```
<?xml version="1.0"?>
<OpenDriveAssetData>
  <Objects>
    <Object>
      <Type> OpenDRIVE type </Type> (Required)
      <Id> OpenDRIVE object "id" </Id> (Optional - only used for Import)
      <Name> OpenDRIVE object "name" </Name> (Optional - only used for Import)
      <Radius> OpenDRIVE object "radius" </Radius> (Optional - only used for Import)
      <Height> OpenDRIVE object "height" </Height > (Optional - only used for Import)
      <FilePath> Relative Asset file path to RoadRunner asset </FilePath> (Required)
    </Object>
  </Objects>
  <Markings>
```

```

    <RoadMark>
      <Type> OpenDRIVE "type" </Type> (Required)
      <Color> OpenDRIVE "color" </Color> (Optional)
      <FilePath> Relative Asset file path to RoadRunner asset </FilePath> (Required)
    </RoadMark>
  </Markings>
  <Signals>
    <Signal>
      <Type> OpenDRIVE "type" </Type> (Required)
      <SubType> OpenDRIVE "subtype" </SubType> (Optional)
      <Id> OpenDRIVE signal "id" </Id> (Optional - only used for Import)
      <Name> OpenDRIVE signal "name" </Name> (Optional - only used for Import)
      <Country> OpenDRIVE signal "country" </Country> (Optional - only used for Import)
      <Value> OpenDRIVE signal "value" </Value> (Optional - only used for Import)
      <FilePath> File path to RoadRunner asset </FilePath> (Required)
      <Variant> Variant of RoadRunner signal/sign asset (integer, where 0 is the first variant, 1 is the second, et
    </Signal>
  </Signals>
</OpenDriveAssetData>

```

Limitations

Here is a list of features that are not exported from RoadRunner to OpenDRIVE.

- Prop polygons (these polygons can be exported only if you run the 'bake' operation to convert them to points). For more details, see **Prop Polygon Tool**.
- Lateral profile (overall road banking is exported, but the full cross-section profile described in the **Cross Section Tool** is not exported)
- Lane height
- Road surface heightfield (CRG)

See Also

External Websites

- ASAM OpenDRIVE

Left-Hand Drive Export to OpenDRIVE 1.4

Recommended Approach

If the scene being built in RoadRunner is meant to have left-hand driving, the `Driving Side` should be set to `Left`. Otherwise, it should be set to `Right`.

OpenDRIVE Details

OpenDRIVE 1.4 does not have a notion of lane travel direction. Instead, it is expected that all drivable lanes on one side of the road go one way and the drivable lanes on the other side of the road go the opposite way.

OpenDRIVE 1.4 does not have a notion of "driving side" (for example, left-hand driving in the UK or Japan). Instead, it is expected that the travel direction be one of these options:

- Assumed right-side (common)
- Assumed based on `<header>` country code (uncommon)
- Determined using the `<incoming>` lanes in `<junction>` entries (uncommon, difficult, and sometimes impossible if no junctions are present)
- Determined by the initial orientation of placed vehicles in a scenario (most common)

RoadRunner Export

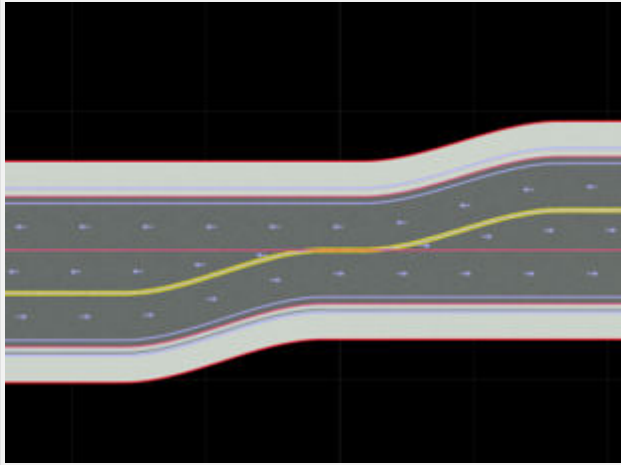
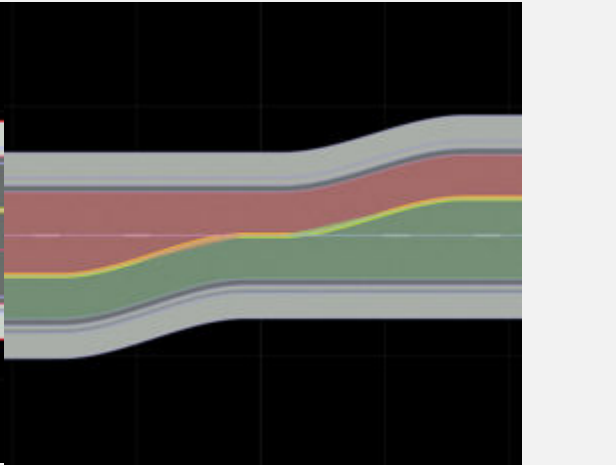
RoadRunner does the following on export for travel direction:

- 1 Ensures that lanes are placed on one side or the other of the OpenDRIVE road based on travel direction
- 2 Writes out the travel direction of the lane in `<userData>` for each lane.

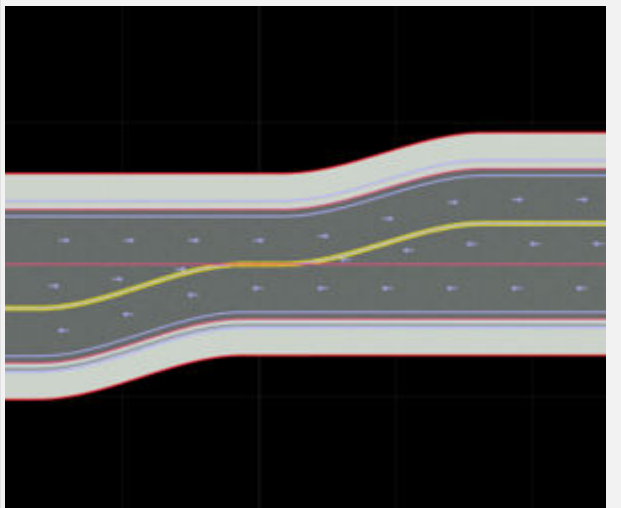
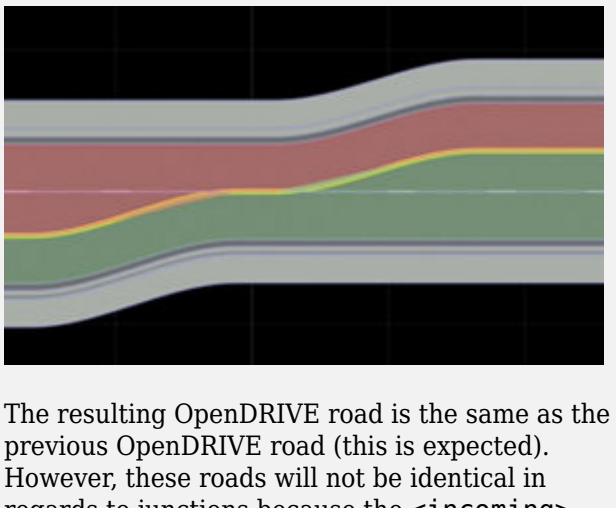
During export, lanes traveling in one direction are placed on one side of the OpenDRIVE road (regardless of the lane's original side of the road in RoadRunner), and the lanes traveling in the opposite direction are placed on the other side of the OpenDRIVE road. The `Driving Side = Left` option provides a hint to the exporter that lanes marked as `Forward` travel direction should (in general) be placed on the `Left` side of the OpenDRIVE road.

Examples

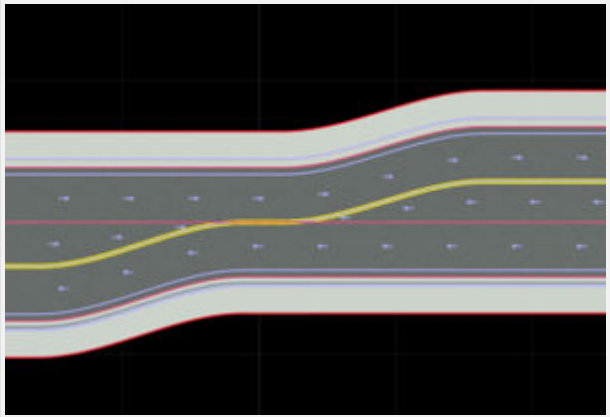
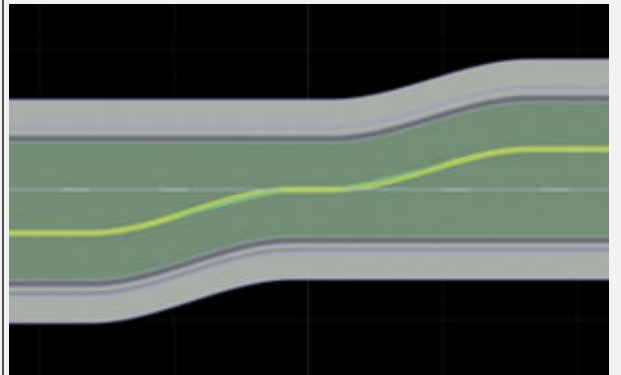
Right-Hand Driving with "Driving Side = Right"

| RoadRunner Scene | Exported OpenDRIVE Scene |
|---|---|
|  |  |
| <p>The forming lanes are created on either side of the center lane. Travel direction is irrespective of the side of the road.</p> | <p>The lanes are placed on each side (colored red and green for left and right, respectively) to adhere to the OpenDRIVE travel direction restrictions.</p> |

Left-Hand Driving with "Driving Side = Left"

| RoadRunner Scene | Exported OpenDRIVE Scene |
|---|--|
|  |  |
| <p>The forming lanes are created on either side of the center lane. Travel direction is irrespective of the side of the road.</p> | <p>The resulting OpenDRIVE road is the same as the previous OpenDRIVE road (this is expected). However, these roads will not be identical in regards to junctions because the <incoming> lanes will be different for the right vs. left cases.</p> |

Left-Hand Driving with "Driving Side = Right" (Not Recommended)

| RoadRunner Scene | Exported OpenDRIVE Scene |
|---|--|
|  |  |
| <p>The forming lanes are created on either side of the center lane. Travel direction is irrespective of the side of the road.</p> | <p>All lanes are on the same side of the road. The following warnings are printed in the Output pane:</p> <p>WARNING: Detected non-critical validation issues.</p> |

Segmentation

Segmentation Overview

RoadRunner has the ability to export scene geometry by category for easy generation of segmentation training data.

Segmentation categories are identified in the export scene by using suffixes applied to each material name. For example, a material named `Concrete` that is applied to a curb is named `Concrete_Curb` on export. The exported material inherits the built-in segmentation category suffix `Curb`, which uniquely distinguishes it from the material `Concrete_Sidewalk` applied to neighboring sidewalk geometry.

Toggle Segmentation Display

- 1 To enter segmentation display, select **View > Sensor > Segmented**.
- 2 To exit segmentation display, select **View > Sensor > Camera**.

Categories

RoadRunner uses a default set of segmentation categories when building geometry within a scene. These categories include: `Road`, `Sidewalk`, `Curb`, `Gutter`, `Marking`, `Ground`, `Building`, `Vehicle`, `Bike`, `Pedestrian`, `Sign`, `Signal`, `Foliage`, and `Prop`.

Additionally, you can extend these categories and assign them to the following project asset types: `Props`, `Signals`, `Lane Markings`, `Polygon Markings`, and `Stencils`.

Add a Custom Category

- 1 In a text editor, open the `SegmentationCategories.xml` file located in the `Project` folder of the project. If the file does not exist, create one.
- 2 Add a new `Category` entry. Include a name and color attribute, which are used during export and segmentation display, respectively.
- 3 Existing categories can also be modified or removed. Changing the name of an existing category is equivalent to removing the old category and adding a new one. Existing assets referencing this old name will default.
- 4 Save the file and restart RoadRunner. New categories are available only after the project is reloaded.

This code shows an example `SegmentationCategories.xml` file.

```
<?xml version="1.0"?>
<CustomSegmentationCategories>
  <Category name="Bush" color="#7BA269"/>
  <Category name="Tree" color="#0F5F32"/>
  <Category name="Crosswalk" color="#963"/>
  <Category name="DashedMarking" color="#369"/>
  <Category name="SolidMarking" color="#48a"/>
  <Category name="DoubleMarking" color="#69b"/>
</CustomSegmentationCategories>
```

Export Scene Geometry Grouped for Segmentation

On export, RoadRunner supports grouping materials by segmentation category or separating them into individual meshes. To toggle between these options, follow these steps.

- 1 Select **File > Export** and select a triangulated format, such as Filmbox or OpenFlight.
- 2 In the export dialog box, fill out the file location and any tiling options.
- 3 Optionally use the **Split by Segmentation** toggle in the **Options** group to control whether each mesh is split by category or remains grouped.
- 4 Click **Export**.

Assign a Category to an Asset

- 1 Select the asset in the **Library Browser**.
- 2 In the **Attributes** pane, under **Segmentation**, select the appropriate category.
- 3 Select **File > Save Project** in the menu bar.

Downloading Plugins

RoadRunner provides plugins for exporting scenes to Unity, Unreal, and CARLA.

The latest plugins can be downloaded [here](#).

The plugins are delivered as a zip file containing subfolders for each plugin type. Refer to the following sections for details on installing and using a specific plugin.

Unity

See “Export to Unity” on page 5-37 for instructions on installing and using the plugin.

Unreal and CARLA

The basic RoadRunner importer and the CARLA integration plugin are included in the same plugin folder. If you are using Unreal and not CARLA, copy only the `RoadRunnerImporter` folder.

See “Export to Unreal” on page 5-54 or “Export to CARLA” on page 5-62 for instructions on installing and using the plugin.

RoadRunner Metadata Export

RoadRunner includes an extra metadata file for certain export options.

Metadata Overview

When exporting to Unity on page 5-37, Unreal on page 5-54, or CARLA on page 5-62, an additional ".rrdata.xml" file is generated during export. This file is used in combination with the RoadRunner import plugins to help cover the information not available in the FBX file. The metadata file holds information about the materials included in the scene and holds traffic signal information. For examples on how to parse this information, refer to the Unity or Unreal plugins included with the RoadRunner installation under the "Tools" folder.

File Details

The metadata file continues to update as needed. The metadata version is stored under the top-level element (for example, <RoadRunnerMetadata Version="3">).

The data is organized into three main sections: SignalConfigurations, Signalization, and MaterialList.

SignalConfigurations

This section holds information about how the signal bulbs change for each configuration of a traffic light (for example, which bulbs are on and off during a green light or red light).

Example:

```
<Signal>
  <ID>{9b15662e-0dae-40d5-ab82-55e0077bcbc2}</ID> // GUID of the signal asset
  <Type>Straight Right</Type> // Supported turn types
  <Configuration> // Configuration entry
    <Name>Red</Name> // State name
    <LightState> // Light bulb mesh state
      <Name>light_red_on</Name> // Name of the mesh node in the signal's FBX
      <State>true</State> // "true" if mesh should be visible
    </LightState>
    <LightState>
      <Name>light_red_off</Name> // Light bulbs typically have a corresponding mesh node for when the
      <State>>false</State>
    </LightState>
    ...
  </Configuration>
  ...
</Signal>
```

Signalization

This section holds information about each traffic junction in the scene and how each signal changes over time.

Example:

```
<Junction>
  <ID>{5c348c08-d2d7-423e-b560-04eb52ddcd10}</ID> // GUID of the junction
  <SignalPhase> // Each signal phase holds a list of
    <Interval> // Each interval represents the state
      <Time>20</Time> // Duration of the interval
    <Signal>
      <ID>{d33d9030-c427-44ff-860b-486f3caf45b2}</ID> // GUID of the signal prop. This can
      <SignalAsset>{9b15662e-0dae-40d5-ab82-55e0077bcbc2}</SignalAsset> // GUID of the signal asset. This can
```

```

        <ConfigurationIndex>2</ConfigurationIndex> // Index into the list of configurat
    </Signal>
    <Signal>
        <ID>{00dd1cc3-9b68-44dd-bb20-a3e49452606f}</ID> // All signals attached to the junct
        <SignalAsset>{9b15662e-0dae-40d5-ab82-55e0077bcbc2}</SignalAsset>
        <ConfigurationIndex>0</ConfigurationIndex>
    </Signal>
    ...
</Interval>
...
</SignalPhase>
...
</Junction>

```

MaterialList

This section contains a list of all the materials used in the scene, along with all the parameters so that they can be reconstructed in the target software.

Example:

```

<Material>
  <Name>Asphalt1</Name> // Name of the material, matches the one stored in the F
  <DiffuseMap>Asphalt1_Diff.png</DiffuseMap>
  <NormalMap>Asphalt1_Norm.png</NormalMap>
  <SpecularMap>Asphalt1_Spec.png</SpecularMap>
  <AmbientColor>1.000000,1.000000,1.000000</AmbientColor> // Ambient color matches diffuse
  <DiffuseColor>1.000000,1.000000,1.000000</DiffuseColor>
  <SpecularColor>0.058824,0.058824,0.058824</SpecularColor>
  <Roughness>0.150000</Roughness>
  <SpecularFactor>1.000000</SpecularFactor>
  <TransparencyFactor>0.000000</TransparencyFactor> // Inverse of diffuse color alpha
  <Emission>0.000000</Emission>
  <TextureScaleU>0.35</TextureScaleU>
  <TextureScaleV>0.35</TextureScaleV>
  <TwoSided>>false</TwoSided>
  <DrawQueue>0</DrawQueue> // Render order for overlapping transparent markings
  <ShadowCaster>>true</ShadowCaster>
  <IsDecal>>false</IsDecal> // Set to "true" for transparent markings
  <SegmentationType>Road</SegmentationType>
</Material>

```

Export to Apollo

Apollo Overview

RoadRunner can export road scenes to Baidu Apollo formats. You can export to Apollo 3.0 and 5.0 XML formats and Apollo 5.0 binary format.

Before you export to any Apollo format, ensure that your scene's world origin is set in the **World Settings Tool**. Once you are ready to export, navigate to **File > Export > Apollo (.bin, .xml)** to open the export options window. Make sure you choose the appropriate Apollo version before completing your export, and ensure that you specify the file extension (.xml or .bin) you intend to export to. Exports to the binary format will include a human readable .txt representation of the protobuf data serialized in the .bin counterpart.

About the Different Apollo Maps

The Apollo Dreamview front-end tool can visualize and simulate routing on RoadRunner Apollo exports. A complete map is composed of the following files for proper simulation:

- `base_map.bin` — A protobuf representation of HD Map information. The representation might be accompanied with a human-readable `.txt` version.
- `sim_map.bin` — A downscaled version of `base_map.bin` used for faster visualization at runtime. The file might be accompanied with a `.txt` version.
- `routing_map.bin` — Topological map information used for generating routes.
- `default_end_way_point.txt` — A start point for routing.

Given either a binary or XML export from RoadRunner, these files can be generated using various tools provided by the Apollo codebase.

Generating Necessary Map Files

Note If you have not yet set up your Apollo Docker[®] environment, follow the Apollo 3.0 guide or Apollo 5.0 guide to do so.

Binary maps can be generated from XML by using the following command in the Apollo Docker environment:

```
bazel-bin/modules/maps/tools/proto_map_generator --map_dir=INPUT_DIR
--output_dir=OUTPUT_DIR
```

INPUT_DIR is the name of the directory containing the XML file, and OUTPUT_DIR is the desired output directory. Within the input directory, ensure the XML file is named `base_map.xml` before running it. This will generate a binary file named `base_map.bin` and a text file version named `base_map.txt` in the output directory specified.

With this .bin file or a .bin file exported directly from RoadRunner, a `sim_map` can be generated with the following command:

```
bazel-bin/modules/maps/tools/sim_map_generator --map_dir=INPUT_DIR
--output_dir=OUTPUT_DIR
```

Again, ensure that the `.bin` file is named `base_map.bin` before running it.

A routing map can be generated with the following command:

```
scripts/generate_routing_topo_graph.sh --map_dir=INPUT_DIR
```

More information about the different Apollo map types can be found [here](#).

Visualizing Maps in Apollo Dreamview

Once you have all of the components for an Apollo map, you can visualize and simulate it in the Dreamview front-end.

Create a folder for your map in `apollo/modules/maps/data`, and add all the map files to that folder. Rename the folder to what you would like to appear in the Dreamview map selection dropdown. Restart Dreamview to refresh the maps in your data folder.

Once Dreamview starts, select your newly added map and a test vehicle in the top-right corner. Ensure that the standard mode is selected.

Go to the **Tasks** tab on the left, and enable **Sim Control** to render the map.

Routing Simulations in Apollo Dreamview

To run a road simulation in Dreamview, ensure that **Routing** is enabled in the **Modules** window. In the routing window, define a route on the map by using at least two waypoints. Click **Send Route Request** to run the simulation.

Visualizing Maps in LGSVL

LGSVL has the ability to import Apollo 5.0 binary files for editing and visualization.

Note If you have not yet set up LGSVL with Unity, follow this guide to do so.

To import an Apollo map into LGSVL, open the **HD Map Import** window under **Simulator > Import HD Map**. Under **Import Format**, select **Apollo 5 HD Map**, and optionally modify the **Distance** and **Delta Threshold** values. Click `...` to open the file browser and select the binary file export. Click **import** to add the map to the scene.

More information about importing maps into LGSVL can be found [here](#).

Apollo User Asset Configuration

The Apollo exporter uses a configuration XML file to map RoadRunner props, signals, signs, and markings to the appropriate `<object>` or `<signal>` "id" and "subtype". This process works the same way as the OpenDRIVE asset configuration.

Exporting a Custom Prop or Signal

- 1 Copy the `ApolloAssetData.xml` file located in the RoadRunner install location under `AssetsInstall/ResourceAssets` to the Project folder in your project (next to the `Project.rrproj` file).
- 2 Open the new `ApolloAssetData.xml` file in a text editor.
- 3 Add entries for new objects, markings, or signals.
- 4 Save your file and export an Apollo file.

You do not need to restart RoadRunner after creating or modifying the `ApolloAssetData.xml` file.

The format of an `ApolloAssetData.xml` file is the same as the format of an `OpenDRIVEAssetData.xml` file. For a detailed definition of this format, see “Export to OpenDRIVE” on page 5-15. The template file in the `AssetsInstall/ResourceAssets` folder also contains examples for a traffic light, stop sign, and yield sign.

Exporting a Traffic Signal with Multiple Variances

Here is the definition of the format to properly set a traffic signal’s asset data for different variances and subsignals.

```
<Signals>
  <Signal>
    <Type>trafficLight</Type>
    <SubType>mix3Vertical</SubType>
    <SubSignals>
      <Variant>1</Variant>
      <SubSignal>
        <LightName>light_red</LightName>
        <Type>circle</Type>
      </SubSignal>
      <SubSignal>
        <LightName>light_yellow</LightName>
        <Type>circle</Type>
      </SubSignal>
      <SubSignal>
        <LightName>light_green</LightName>
        <Type>circle</Type>
      </SubSignal>
    </SubSignals>
    <SubSignals>
      <Variant>2</Variant>
      <SubSignal>
        <LightName>light_red</LightName>
        <Type>arrowLeft</Type>
      </SubSignal>
      <SubSignal>
        <LightName>light_yellow</LightName>
        <Type>arrowLeft</Type>
      </SubSignal>
      <SubSignal>
        <LightName>light_green</LightName>
        <Type>arrowLeft</Type>
      </SubSignal>
    </SubSignals>
  </SubSignals>
</Signals>
```

```
<Variant>3</Variant>
<SubSignal>
  <LightName>light_red</LightName>
  <Type>arrowRight</Type>
</SubSignal>
<SubSignal>
  <LightName>light_yellow</LightName>
  <Type>arrowRight</Type>
</SubSignal>
<SubSignal>
  <LightName>light_green</LightName>
  <Type>arrowRight</Type>
</SubSignal>
</SubSignals>
<FilePath>Props/Signals/Signal_3Light_Post01.fbx</FilePath>
</Signal>
</Signals>
```

Unsupported Features

The following data records are currently unsupported for export.

- Route View Record (<routeView> ... </routeView> and its children)
- Lane Overlap Group (<laneOverlapGroup> ... </laneOverlapGroup> and its children)

See Also

Export to Metamoto

RoadRunner can export road scenes for use in Metamoto simulations. To export to Metamoto, follow these steps:

- 1 Select **File > Export > Metamoto**.
- 2 Specify a file name. All other export options are set for exporting to Metamoto.

When you export, the software generates a zip file containing the files needed to use your scene in Metamoto. These zipped files have the same name you specified when exporting. For example, *MyScene.zip* would contain:

- *MyScene.fbx* (and any other necessary texture files)
- *MyScene.xodr*
- *MyScene.geojson*
- *MyScene.rrdata.xml*

For any limitations, refer to the documentation about exporting to FBX on page 5-3, OpenDRIVE® on page 5-15, and GeoJSON on page 5-9.

See Also

External Websites

- <https://www.metamoto.com/>

Export to Unity

Unity Overview

RoadRunner can export scenes to Unity format. The Unity export option exports a Filmbox (.fbx) file containing the 3D objects in a scene along with an additional XML file to hold extra data for materials and traffic signals in the scene.

On the Unity side, a set of scripts are included in the RoadRunnerUnityTool asset package to help import the FBX file using the information stored in the XML file. The script handles the following details:

- Setting up materials
 - Material data is read in from the XML file and mapped into the included custom shaders.
- Adding colliders to roads and terrain
 - Colliders are added to all imported meshes.
- Setting up the components of traffic signals
 - Signal data is read in from the XML file to create a new game object in the prefab, with the light bulb references to game objects set up during import.
 - The traffic signals will cycle through their phases during play mode.
 - The UUIDs prefixed in the game object for prop instances are needed only at import time to set up references to game objects in the traffic signal script so that they can be renamed freely.
- Unity software requirements: Unity Version 2017.3+

Installing the Import Tool

Follow the instructions in this section to install the Import Tool into your Unity project.

- 1 See the page “Downloading Plugins” on page 5-29 for instructions for downloading the latest version of the plugin.
- 2 Extract the RoadRunner Plugins zip file and locate the "RoadRunnerUnityTool.unitypackage" file in the "Unity" folder.
- 3 Open your project in Unity.
- 4 Open the RoadRunnerUnityTool asset package file to import it. Alternatively, drag the package file into the Unity Project window, or select

Assets > Import Package > Custom Package and then select the package file.

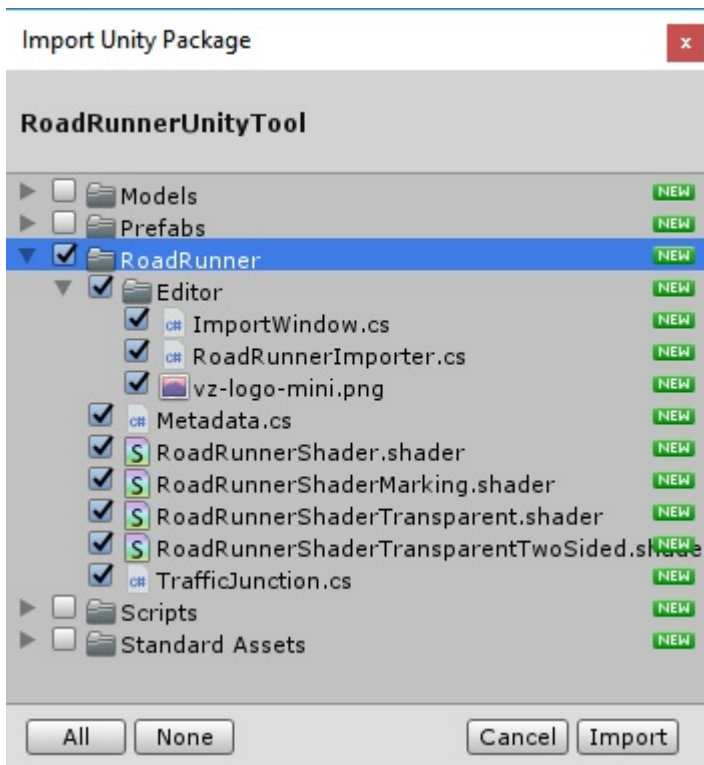
- 5 Click **Import** in the Import Unity Package dialog box.



Selecting Package Files to Import

The package includes some extra files to add a drivable vehicle to your scene. If you do not need these extra files, then you can deselect them when you import the package.

The essential scripts inside the "RoadRunner" folder are needed to set up the materials and traffic signals in the scene.



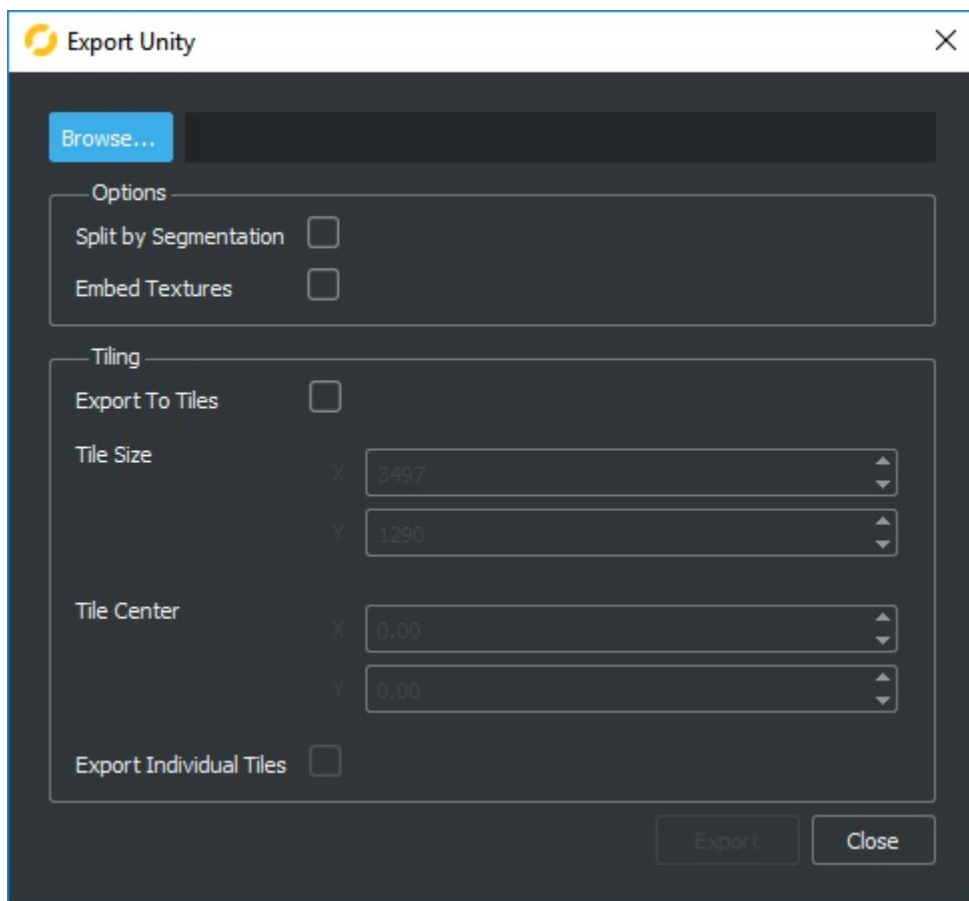
Package Contents

- Models: Mesh, materials, and textures for the drivable car prefab.
- Prefabs: The prefab for the car with scripts set up.
- RoadRunner:
 - ImportWindow.cs: Editor window to display messages for the RoadRunner importer.
 - RoadRunnerImporter.cs: Editor script for importing the FBX file with the data from the XML file.
 - Metadata.cs: Contains classes to hold the imported metadata XML file.
 - TrafficJunction.cs: Component for controlling signals from data in the XML file at import time.
 - Various shaders to match RoadRunner material settings.
- Scripts: For the Sedan prefab.
- Standard Assets: For the Sedan prefab.

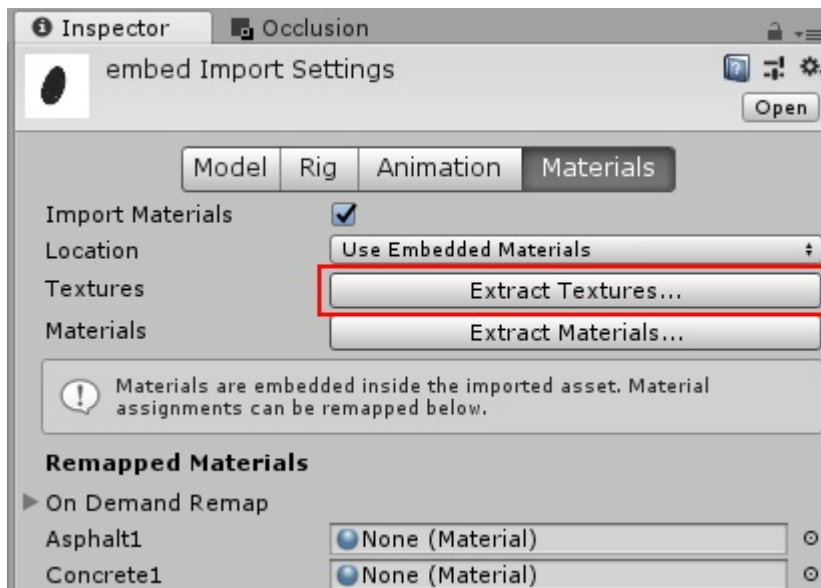
Exporting from RoadRunner to Unity

Follow these steps to export a scene from RoadRunner to Unity:

- 1 Open your scene in RoadRunner.
- 2 Export the scene to Unity format using **File > Export > Unity (.fbx + .xml)** from the menu bar.
- 3 In the Export Unity dialog box, set your desired options, and then click **Export**.



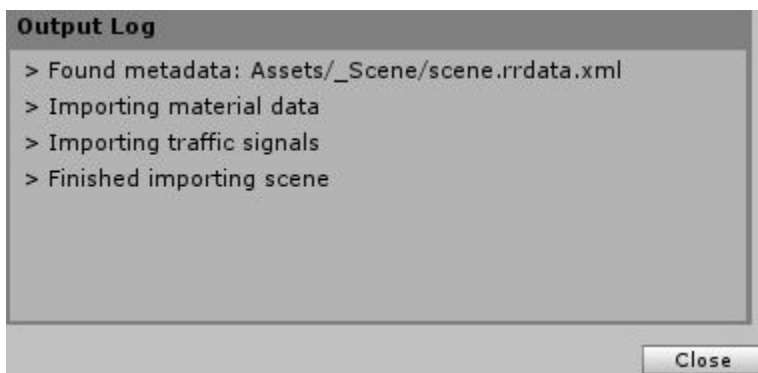
- 4 Browse to open the file dialog box to set the exported file's name and path. The textures and the XML file are exported to the same folder. (Tip: Create a new folder when choosing a file location, so you can import the entire folder into Unity.)
 - The mesh can be split by segmentation type. Meshes have "<segmentation type>Node" appended to their names.
 - If the **Export To Tiles** option is selected, meshes are split per tile. Props are grouped by the tile they are in.
 - By default, only one file is exported. Tiles are stored in separate nodes.
 - If **Export Individual Tiles** is enabled, each tile will be stored in its own FBX file.
 - When exporting with **Embed Textures** selected, you need to manually extract the textures inside Unity.



Importing into Unity

To import a scene into Unity that you previously exported from RoadRunner, drag all the exported files (or the entire folder) into the Unity project window. Alternatively, use **Assets > Import New Asset** in Unity and select all the exported files.

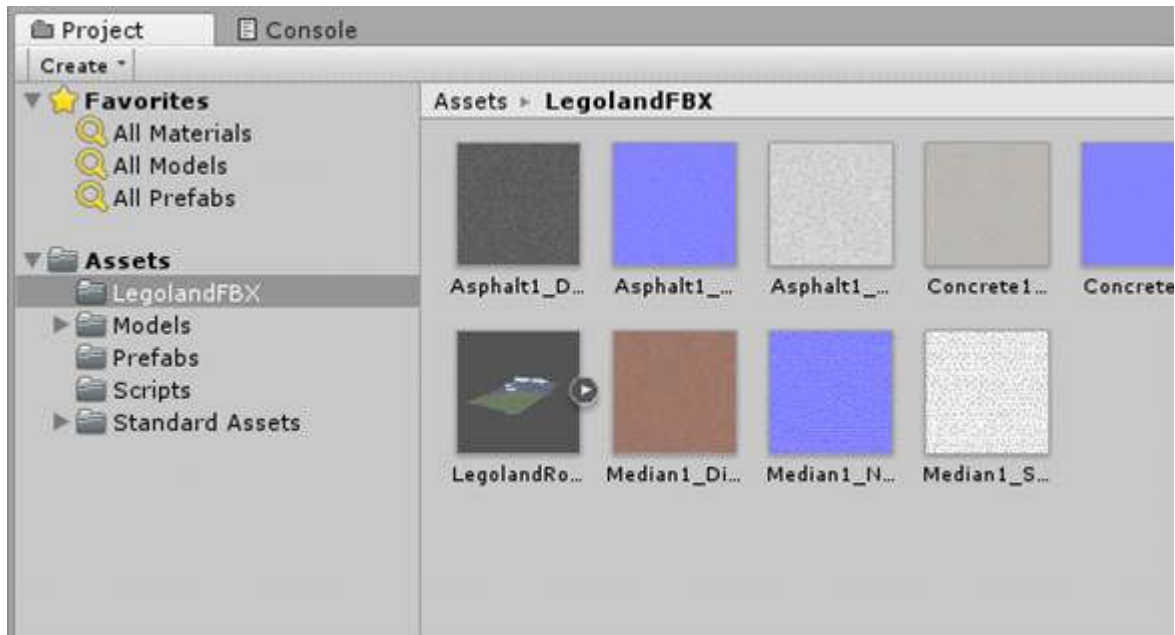
The output window that opens contains log messages from the import plugin.



(Optional) Test Drive in Unity

You can place and drive a car model around an imported scene by following these steps:

- 1 Drag the Sedan prefab from the Prefabs folder into the scene. (Note: In some versions of Unity, you might need to manually tag the Main Camera as "MainCamera" for some scripts to work.)



- 2 Click **Run**.

About Importing Traffic Signals into Unity

If traffic signals were set up in RoadRunner, then they are imported into Unity as junction controllers. These controllers are automatically created during import and attached to the prefab.

Prop instances for traffic signals are prefixed by their UUID so that the traffic signal controller has a way to identify which signals it controls. The TrafficJunction script handles the logic for switching between signal states.

FBX details

The FBX file is identical to the one exported from the Filmbox export option. The only difference is the extra rrdata.xml metadata file.

Setting Up the Sample Vehicle

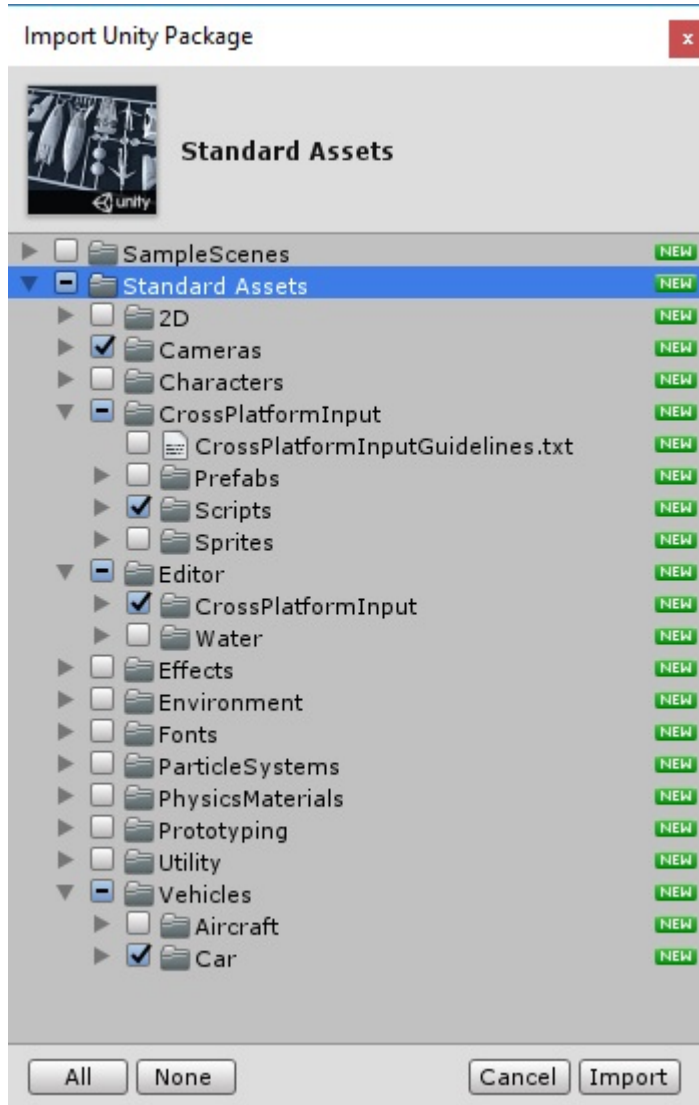
The RoadRunnerUnityTool unitypackage also includes the RoadRunner Sedan 3D model. This section covers how to set it up with Unity Standard Assets.

Note The following section was tested on Unity 2019.1. Older versions might require different steps to modify prefabs.

Adding the Standard Assets

- 1 Download the "Standard Assets (for Unity 2017.3)" package from the Unity Asset Store.
- 2 Select the following folders to import:
 - "Standard Assets/Cameras"

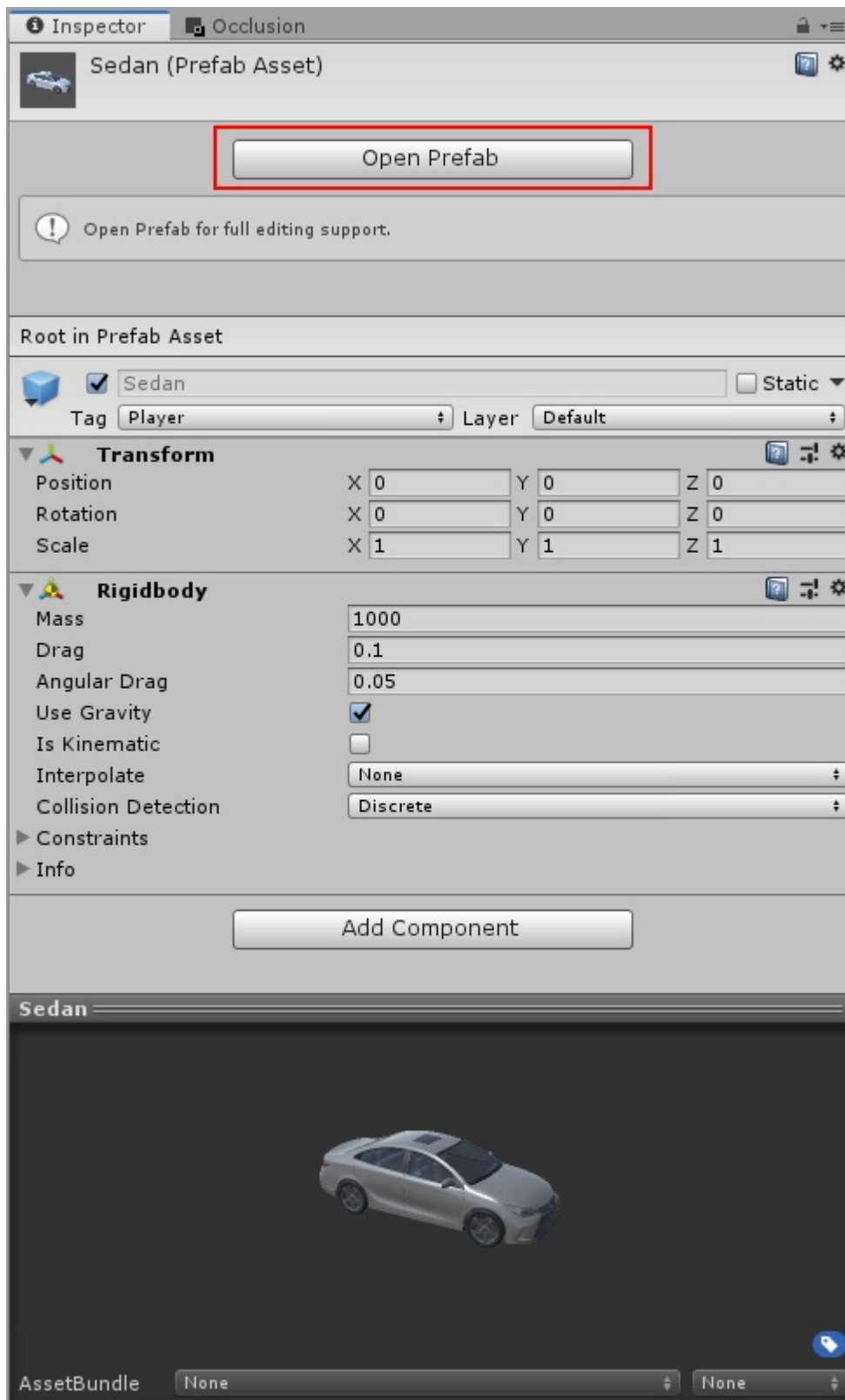
- "Standard Assets/CrossPlatformInput/Scripts"
- "Standard Assets/Editor/CrossPlatformInput"
- "Standard Assets/Vehicles/Car"



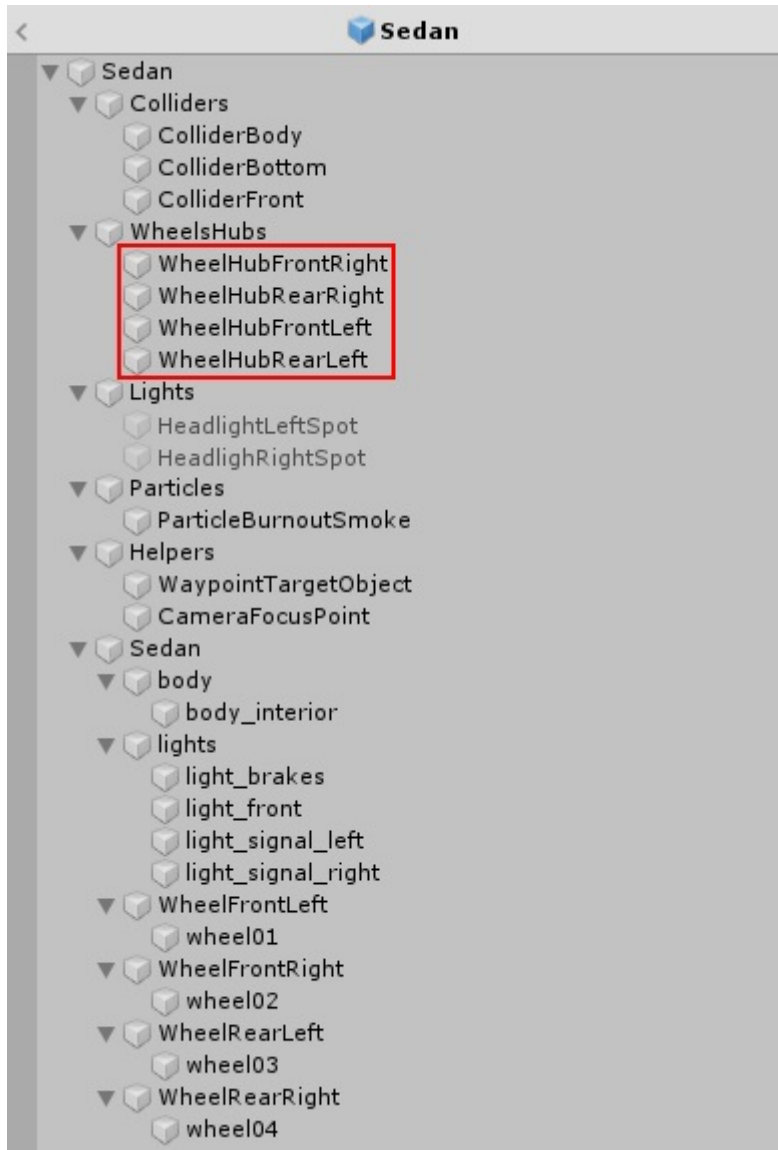
- 3 Import the package by clicking **Import**.

Setting Up the Sedan Prefab

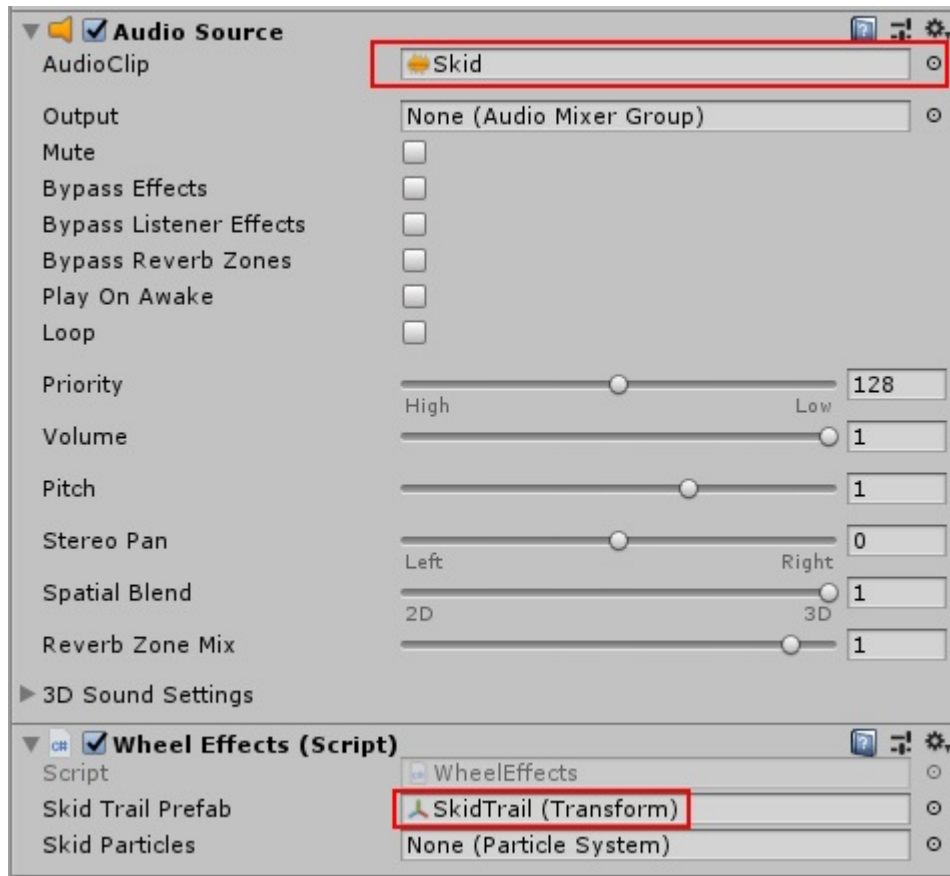
- 1 Select the Sedan prefab (located in "Assets/Prefabs") and click **Open Prefab** in the Inspector window.



- 2 For each Wheel Hub Game Object under "Sedan/WheelsHubs" (for example, "WheelHubFrontRight"), complete these steps:



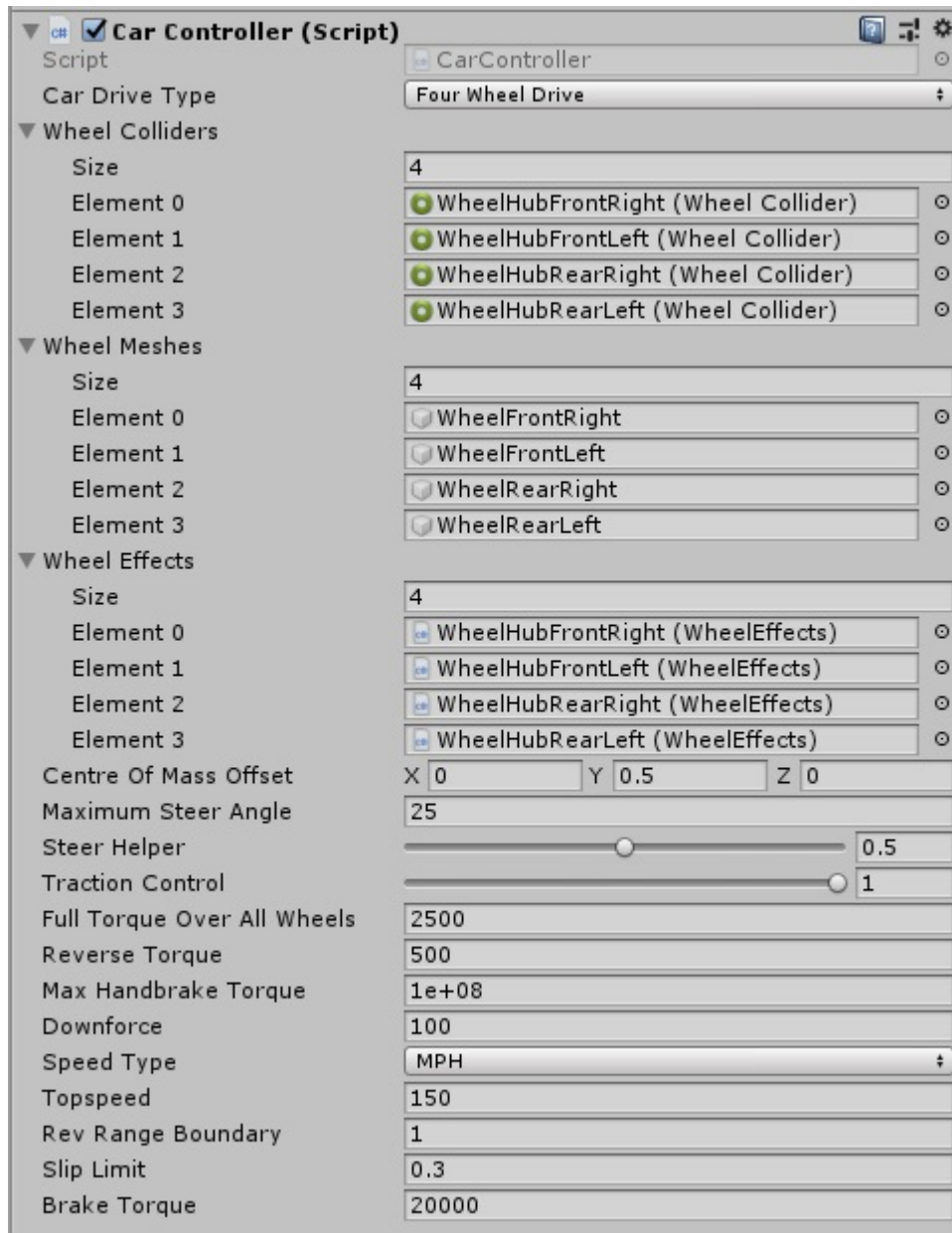
- a Add the "Wheel Effects" component.
- Set the "Skid Trail Prefab" to the "SkidTrail" Prefab (located in "Assets/Standard Assets/Vehicles/Car/Prefabs").
- b Set the Audio Source's "AudioClip" to the "Skid" sound effect if it is missing.



- 3 Select the top level "Sedan" game object in the hierarchy window.

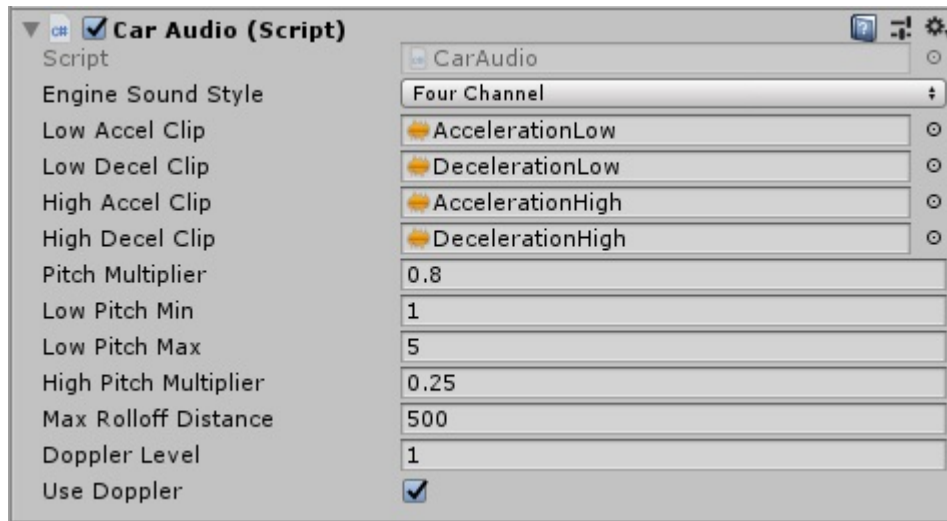


- a Add the "Car Controller" component with the following settings.

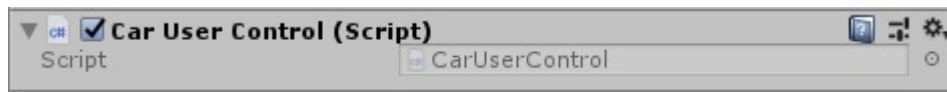


To avoid errors, verify that the order of the wheels is correct.

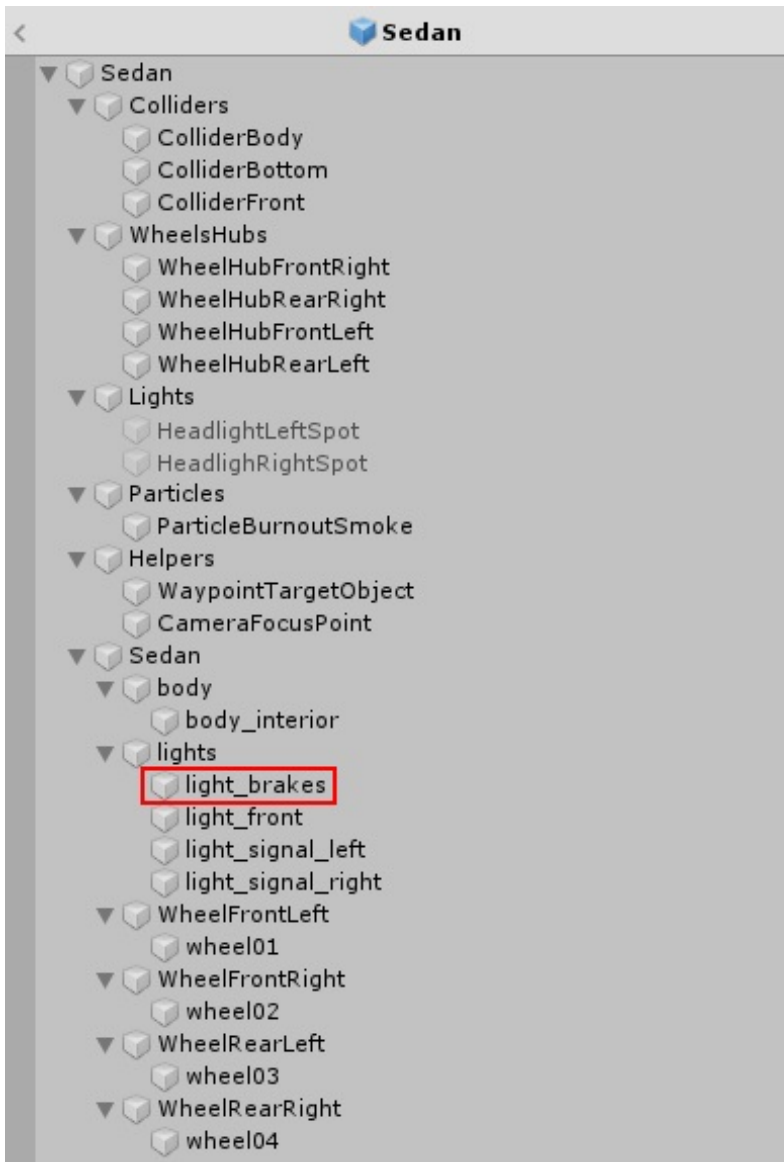
- b** Add the "Car Audio" component with the following settings.



- c Add the "Car User Control" component.



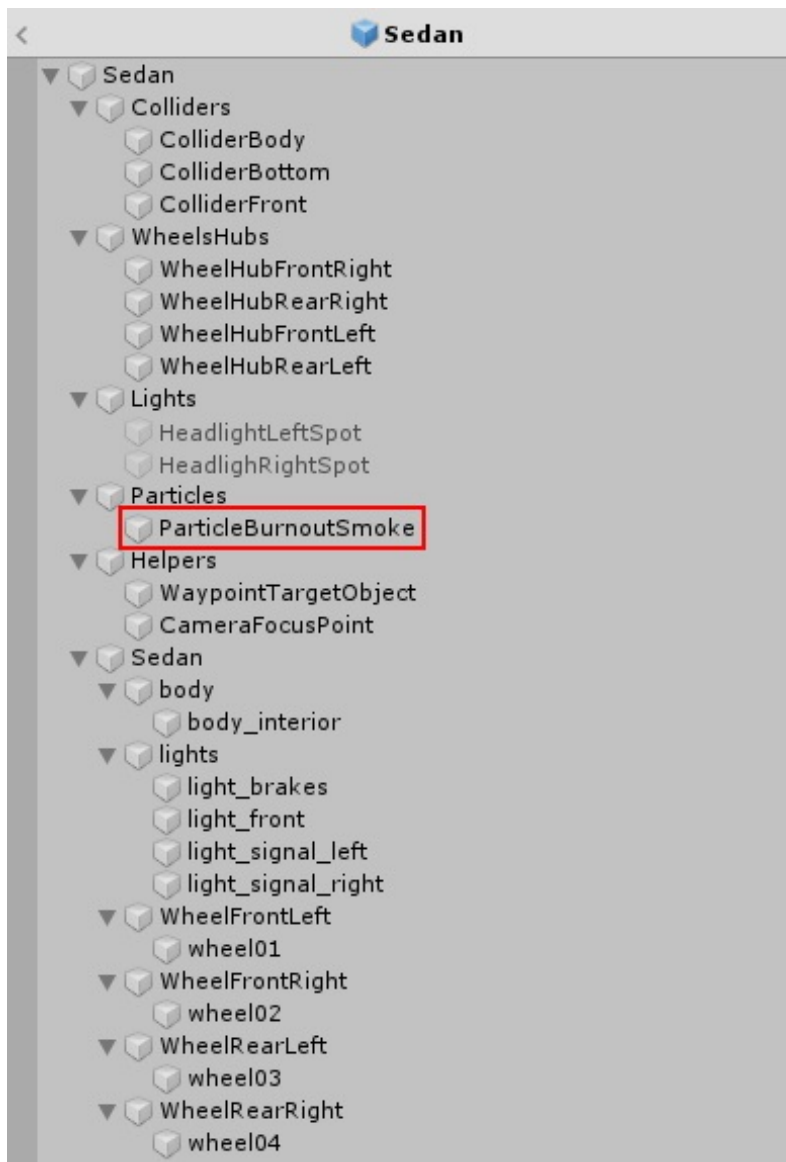
- 4 Select the "light_brakes" game object.



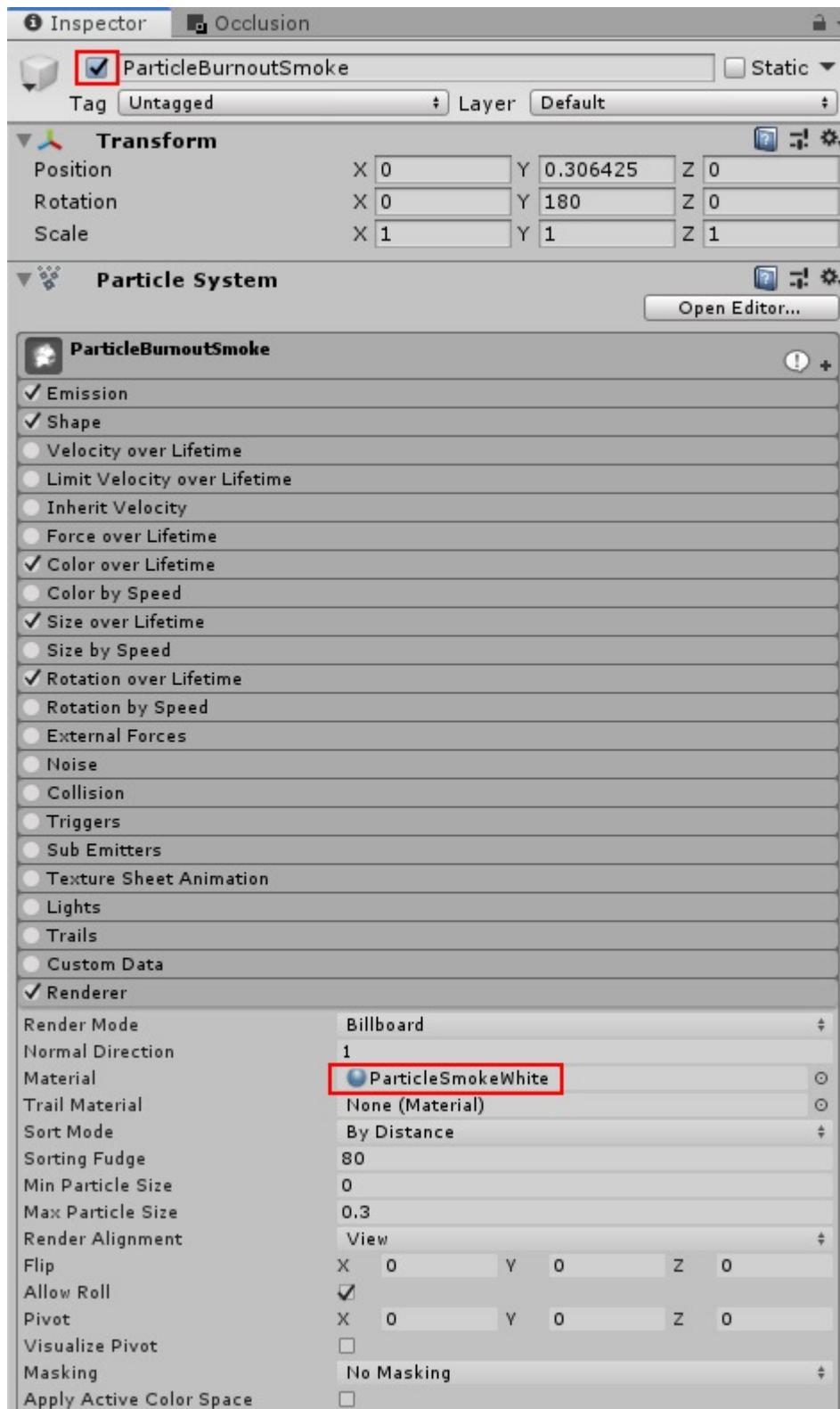
- a Add the "Brake Light" component and set the "Car" property to the top level "Sedan" game object.



- 5 Select the "ParticleBurnoutSmoke" game object.



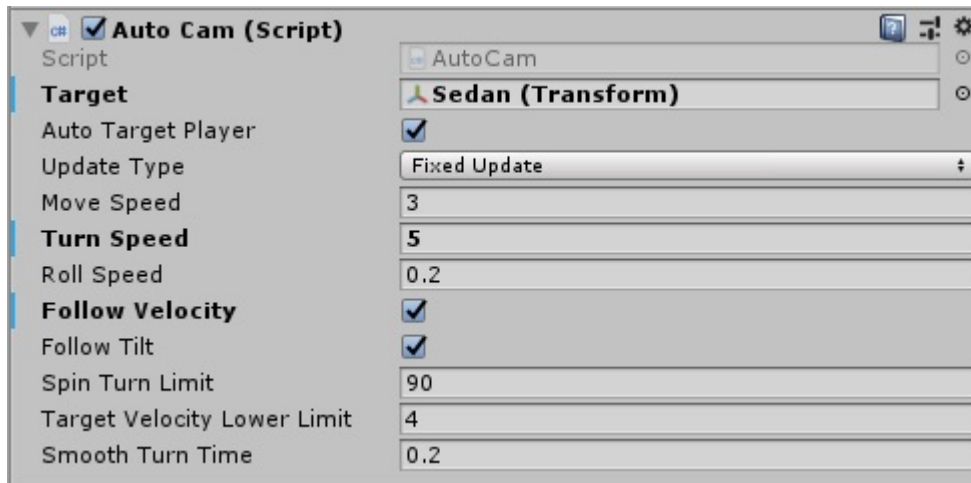
- a Enable the game object.
- b Set the Particle System component's Renderer Material property to "ParticleSmokeWhite".



6 Exit Prefab edit mode.

Setting Up the Camera

- 1 Delete any existing cameras in the scene.
- 2 Add the Sedan prefab to the scene.
- 3 Add the "MultipurposeCameraRig" prefab (located in "Standard Assets/Cameras/Prefabs") to the scene with the following settings.



Running the Scene

Add in an imported RoadRunner scene and click play to drive around in it.



Export to Unreal

Unreal Overview

RoadRunner can export scenes to Unreal. The Unreal export option exports a Filmbox (.fbx) file and generates an additional XML file to hold extra data. The XML file holds data for materials and traffic signals in the scene.

On the Unreal side, a plugin is provided to help import the FBX file by using the information stored in the XML file. The plugin handles the following:

- Setting up materials
 - Material data is read in from the XML file and maps the data into new instance of one of the base materials included with the plugin.
 - Transparent materials will choose between the translucent and masked blend modes based on the transparency of the diffuse color.
- Adjusting the colliders in the imported static meshes
 - During import, newly created static mesh assets have their "Collision Complexity" property set to "Use Complex Collision As Simple".
- Setting up the traffic signal components:
 - Signal data is read in from the XML file and creates a component in the blueprint with the light bulb names set up during import.
 - The traffic signals will cycle through their phases during play mode.
 - The UUIDs prefixed in the scene components for prop instances are needed to reference the static mesh component in the traffic signal script during play mode, so signals will not work if their names are changed.
- Unreal software requirements: Unreal Version 4.17+

Installing the Plugin

Follow the instructions in this section to install the Unreal plugin.

- 1 See "Downloading Plugins" on page 5-29 for instructions for downloading the latest version of the plugin.
- 2 Extract the RoadRunner Plugins zip file and locate the "RoadRunnerImporter" folder and "RoadRunnerMaterials" folders under "Unreal/Plugins".

Note The Unreal plugin folder now also contains a RoadRunnerCarla integration plugin. Do not copy this folder if you are not using CARLA.

- 3 Copy the "RoadRunnerImporter" folder and "RoadRunnerMaterials" folder into the "Plugins" folder under the project directory. If a "Plugins" folder does not exist, create one.

| Name | Type | Size |
|----------------------|-----------------------|------|
| Config | File folder | |
| Content | File folder | |
| Intermediate | File folder | |
| Plugins | File folder | |
| Saved | File folder | |
| packageTest.uproject | Unreal Engine Proj... | 1 KB |

4 Rebuild the plugin.

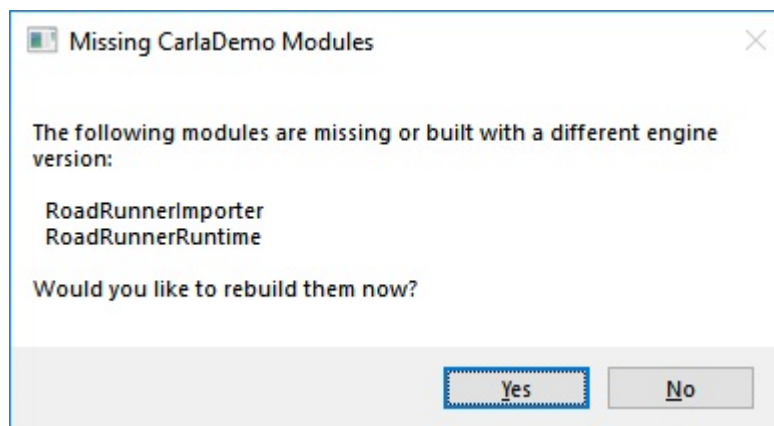
a Generate the project files.

- Windows - Right-click the .uproject file and select "Generate Visual Studio project files."
- Linux - Run this code at the command line:

```
$UE4_ROOT/GenerateProjectFiles.sh -project="<Path to .uproject file>" -game -engine
```

Set UE4_ROOT to your Unreal Engine install directory.

b Open the project and build the plugins by clicking **Yes**.



5 The plugin shows up under **Edit > Plugins**. If it does not appear in that menu, check that the **Enabled** parameter is selected.

Plugin Contents

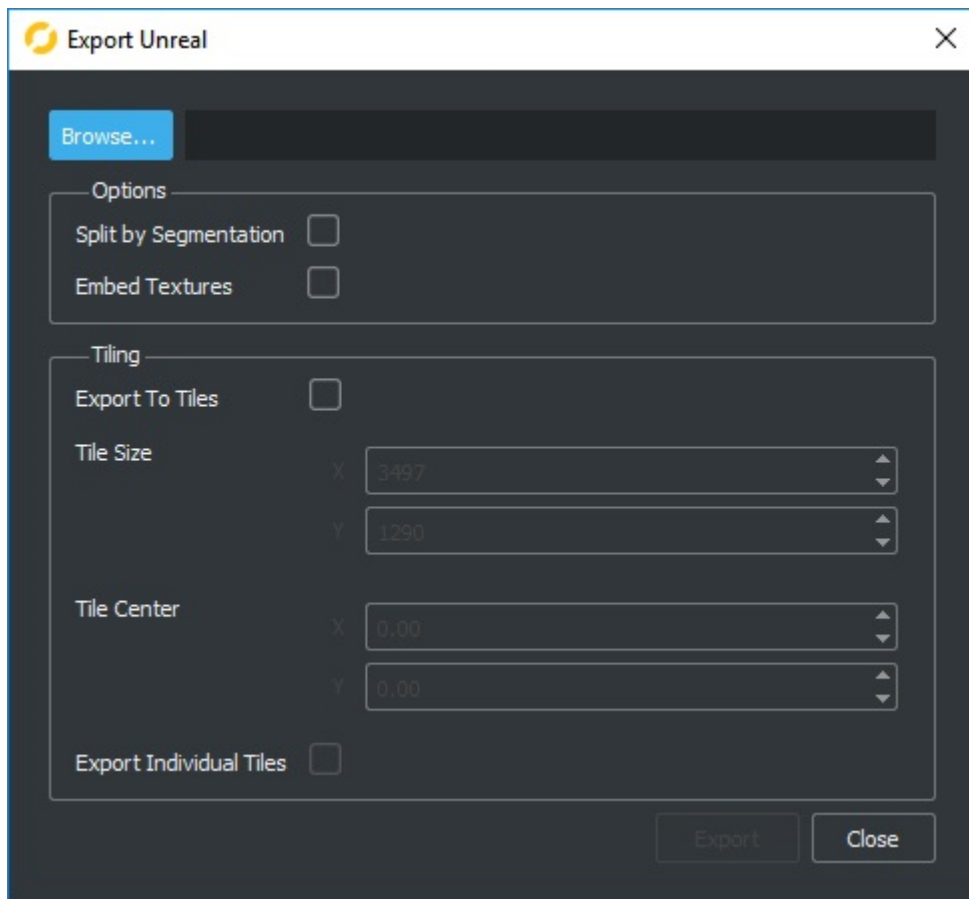
- RoadRunnerImporter module:
 - Overrides the default FBX importer when the metadata file is present
 - Option to overwrite default materials with new materials using the metadata file
 - Import signal data and timing
- RoadRunnerRuntime module:
 - Contains component to control traffic signal visuals
- RoadRunnerMaterials plugin:

- Base materials to create instances from

Exporting from RoadRunner to Unreal

Follow these steps to export a scene from RoadRunner to Unreal:

- 1 Open your scene in RoadRunner.
- 2 Export the scene using the Unreal option. Select **File > Export > Unreal (.fbx + .xml)** from the menu bar.
- 3 In the Export Unreal dialog box, set the mesh merging and tiling options, and then click **Export**.



- 4 Browse to open the file dialog box to set the exported file's name and path. The FBX, textures, and XML files are exported to the same folder.
 - The mesh can be split by segmentation type. Meshes have "<segmentation type>Node" appended to their names.
 - If the **Export To Tiles** option is selected, meshes are split per tile. Props are grouped by the tile they are in.
 - By default, only one file will be exported. Tiles will be stored in separate nodes.
 - If **Export Individual Tiles** is enabled, each tile will be stored in its own FBX file.

Importing into Unreal

There are multiple ways to import the scene into Unreal:

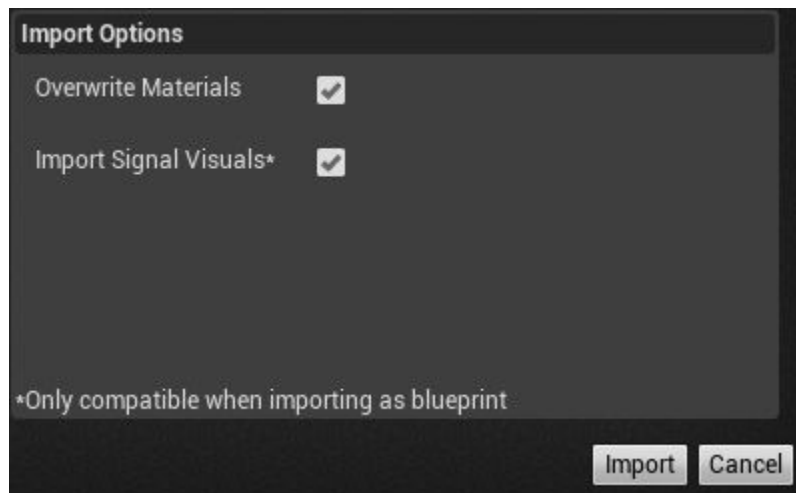
- Drag the file into the Content Browser.
- Use the "Import" button and select the FBX file.

The plugin checks if there is a RoadRunner XML file associated with the imported file and imports as normal if a corresponding XML file is not found.

Selecting **File > Import Into Level** does not use the exported RoadRunner XML and uses the Unreal importer instead.

Prop Instances are prefixed by their UUID so that the traffic signal controller has a way to identify which signals it controls.

When the RoadRunner Import Options Dialog Box Opens



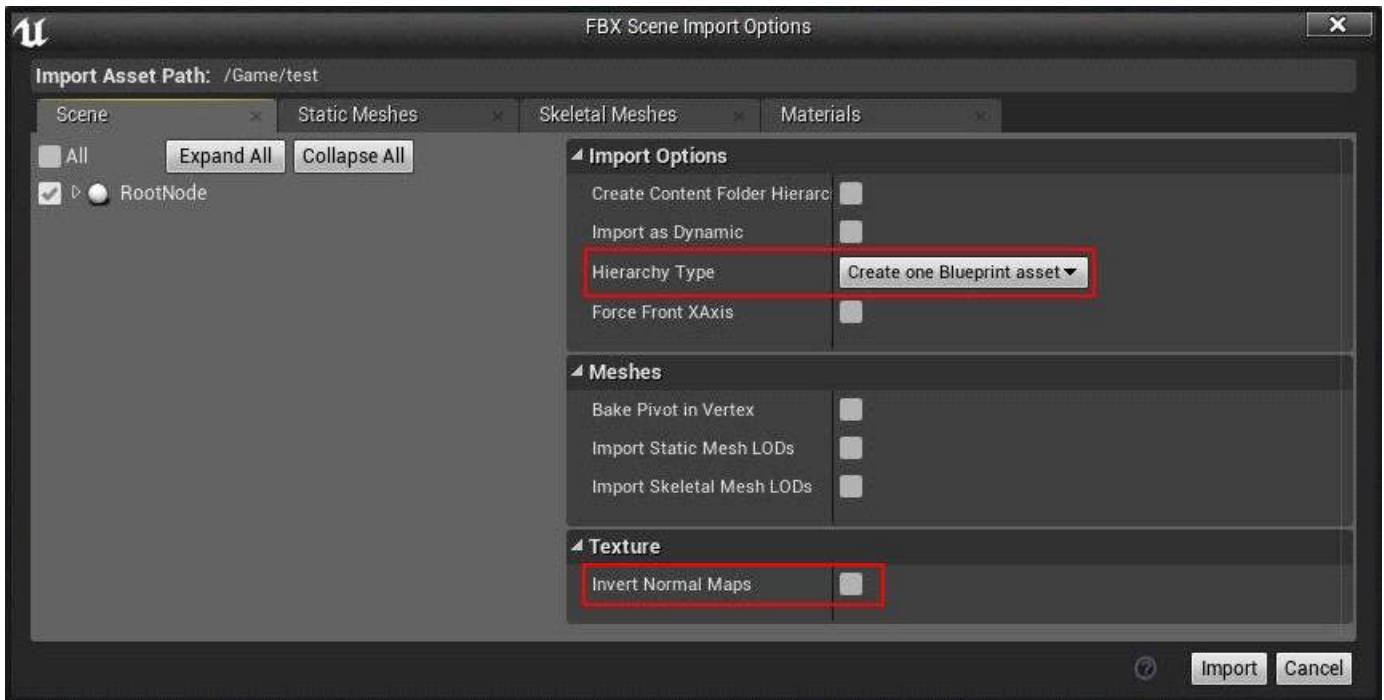
- Overwrite Materials
 - Overrides the default material importing
 - Needs to be unchecked if you want to set the materials to "Use Existing" in the next dialog box
- Import Signal Visuals
 - Functional only when "Create one Blueprint asset" is selected in the next dialog box

When the FBX Scene Import Options Dialog Box Opens:

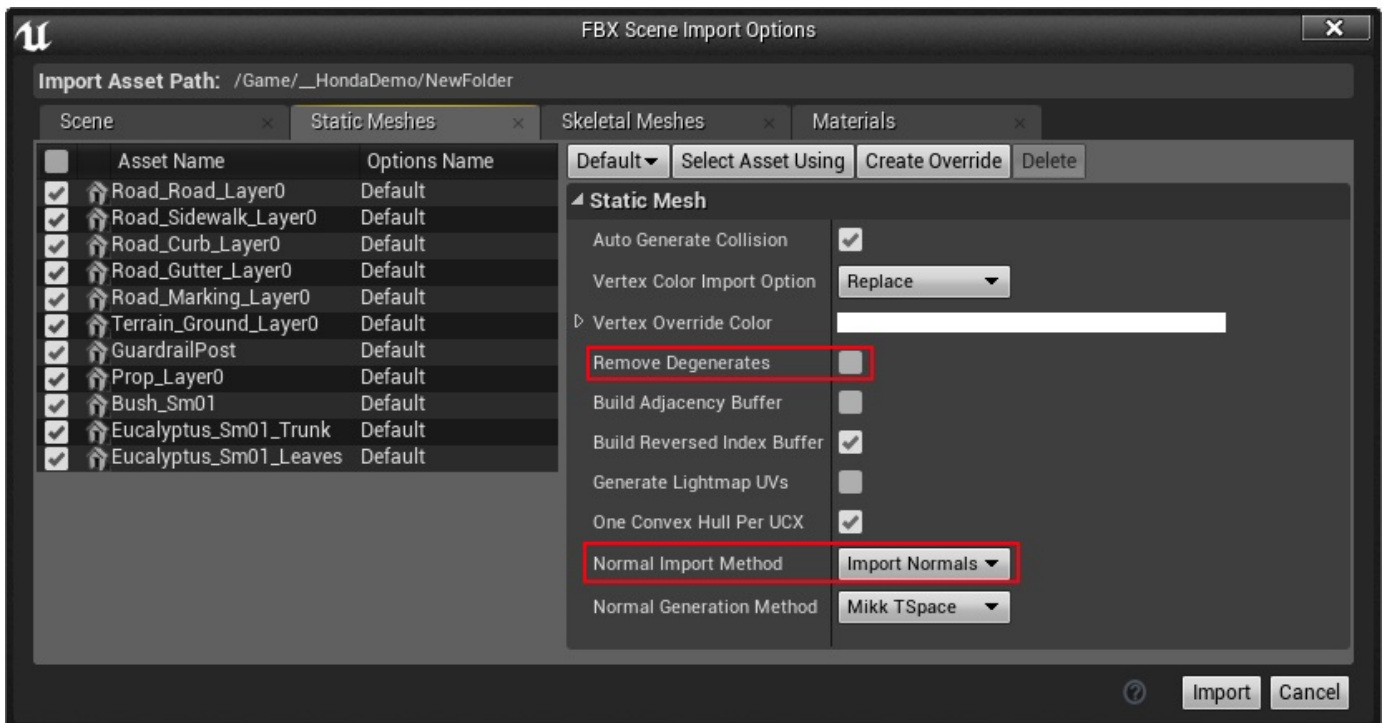
- 1 Set **Scene > Hierarchy Type** to "Create one blueprint asset" (selected by default).

Note Only the "Create one Blueprint asset" import option works with materials, signals, and transparency sorting. The "Create one Actor with Components" and "Create Level Actors" options import only materials.

- 2 Select **Invert Normal Maps** if needed.



- 3 Set **Static Meshes** > **Normal Import Method** to "Import Normals".



- 4 (Optional) Clear the **Remove Degenerates** parameter, which can help for some props created in a larger scale.
- 5 Click **Import**.

About Importing Traffic Signals into Unreal

If traffic signals were set up in RoadRunner, they are imported into Unreal as RoadRunnerTrafficJunction components. These controllers are automatically created during import and included in the created blueprint.

The RoadRunnerTrafficJunction component handles the logic for switching between signal states. UUIDs are used to match to specific game objects in the scene.

FBX Details

The FBX file automatically splits the mesh by transparency sorting layer. This is due to Unreal storing "Translucency Sort Priority" on the static mesh component.

sRGB Textures

Unreal Engine does not support 16-bit sRGB textures. Therefore, textures appear to be washed out, unless the texture files are converted to 8-bit sRGB textures.

Large Scene Optimizations

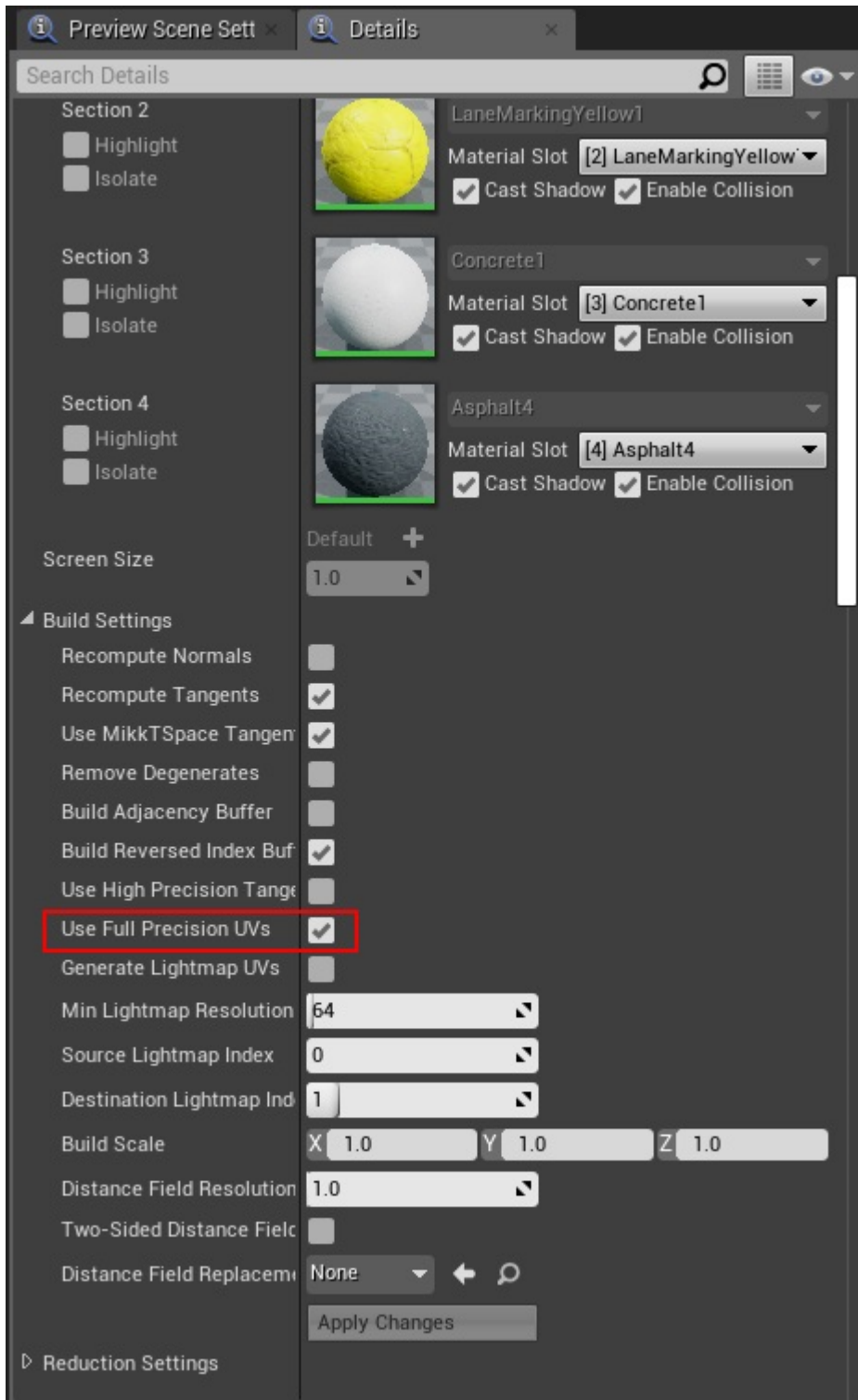
Using the "Create one Actor with Components" option can be more efficient. However, signals will not be set up.

Importing Without the Plugin

This section covers fixes handled automatically by the RoadRunner plugin.

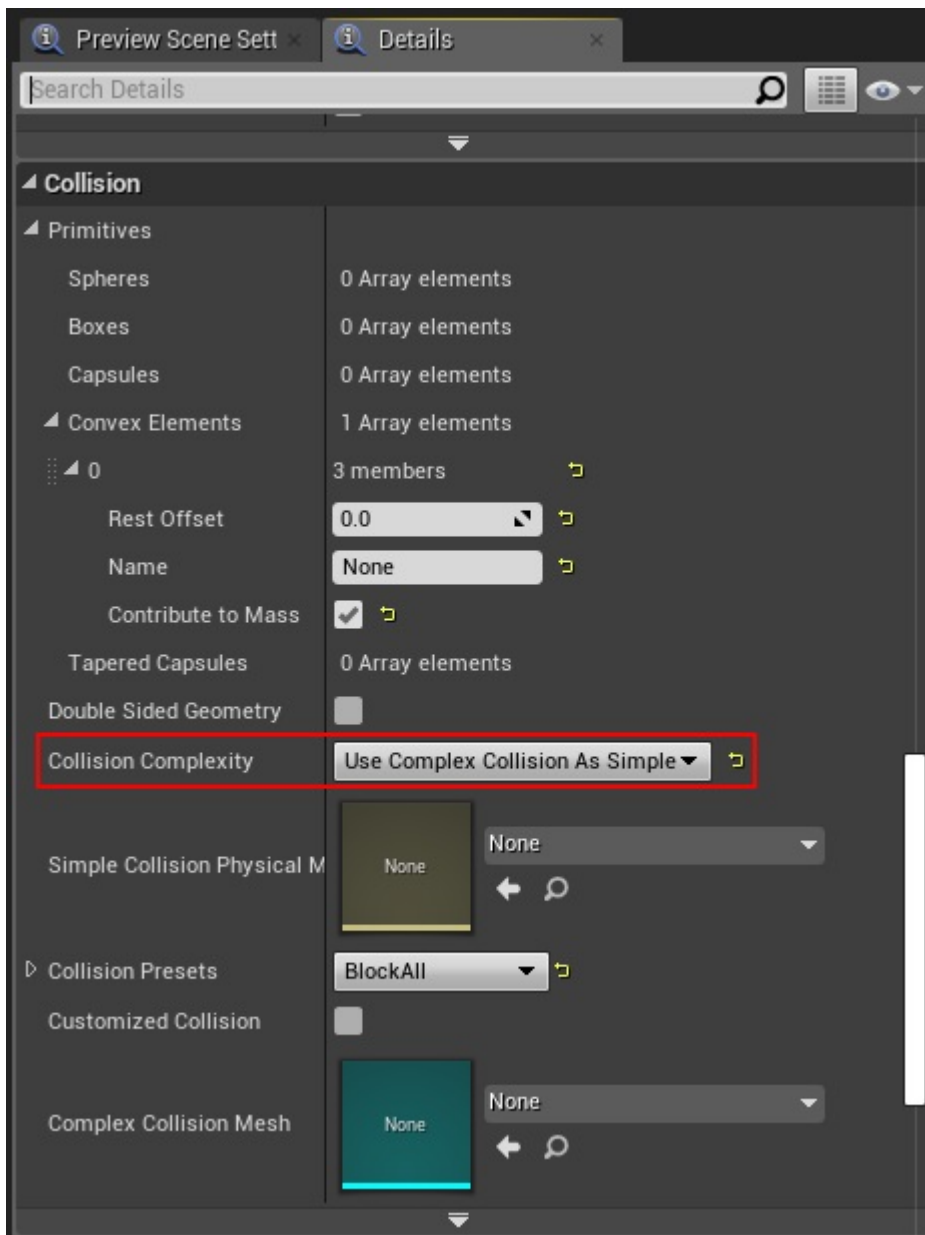
Fix Stretched Textures

Meshes with UV coordinates far away from the 0 to 1 range can cause issues in Unreal. On Static Mesh assets with this issue, the "Use Full Precision UVs" option can be set to fix it.



Fix Objects Floating Above the Road

Static Mesh assets need to have their Collision Complexity property set to "Use Complex Collision As Simple". Otherwise, collision boxes need to be manually added.



Known Issues

In exported scenes, terrain sensors or any other actor that uses the Unreal Engine line tracing API, might not detect hits near the road markings that have nonconvex shapes. To enable these detections, before exporting, float the road markings slightly above the road surface. From the RoadRunner menu, select **Tools > LOD Settings**. Then, in the LOD Settings dialog box, set **Marking Construction Type** to **Floating** and increase **Floating Marking Offset** to approximately 0.005 meters. For more details on floating lane markings, see “Customize Levels of Detail in Exported Scenes” on page 5-77.

Export to CARLA

CARLA Export Overview

RoadRunner can export scenes to the CARLA simulator. The CARLA export option exports a Filmbox (.fbx) file, an XML for some metadata, and an OpenDRIVE (.xodr) file. The XML file holds data for materials in the scene.

On the CARLA or Unreal side, a plugin is provided to help import the FBX file by using the information stored in the XML file. The plugin handles the following:

- Setting up materials
 - Material data is read in from the XML file and maps the data into a new instance of one of the base materials included with the plugin.
 - Certain materials will instantiate from one of the CARLA materials.
 - Transparent materials will choose between the translucent and masked blend modes based on the transparency of the diffuse color.
- Adjusting the colliders in the imported static meshes
 - During import, newly created static mesh assets have their "Collision Complexity" option set to "Use Complex Collision As Simple".
- Setting up the traffic signal visuals
 - Traffic signal logic is not hooked up to the simulator.
- Software requirements
 - Unreal Version 4.21+
 - CARLA 0.9.5 built from source

Installing the Plugins

Follow the instructions in this section to install the Unreal plugin:

- 1 See "Downloading Plugins" on page 5-29 for instructions for downloading the latest version of the plugin.
- 2 Extract the RoadRunner Plugins zip file and locate the "RoadRunnerImporter", "RoadRunnerCarlaIntegration", and "RoadRunnerMaterials" folders under "Unreal/Plugins".
- 3 Copy the "RoadRunnerImporter", "RoadRunnerCarlaIntegration", and "RoadRunnerMaterials" folders into the "Plugins" folder under the CarlaUE4 project directory, located at <carla>/Unreal/CarlaUE4/Plugins (next to the "Carla" folder).



4 Rebuild the plugin.

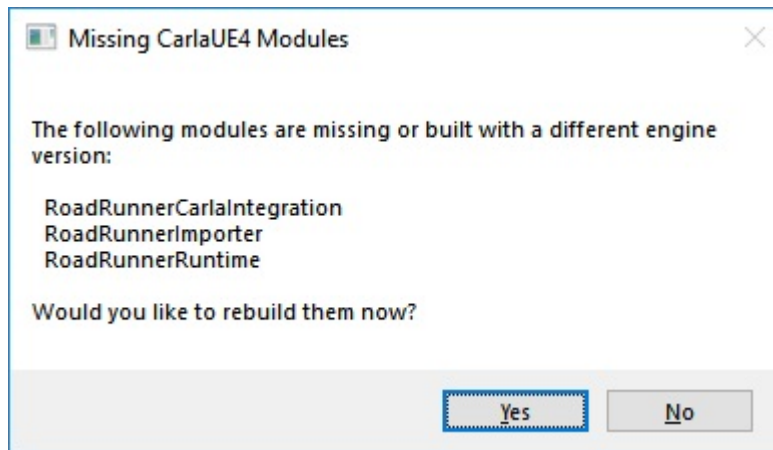
a Generate the project files.

- Windows - Right-click the .uproject file and select "Generate Visual Studio project files."
- Linux - Run this code at the command line:

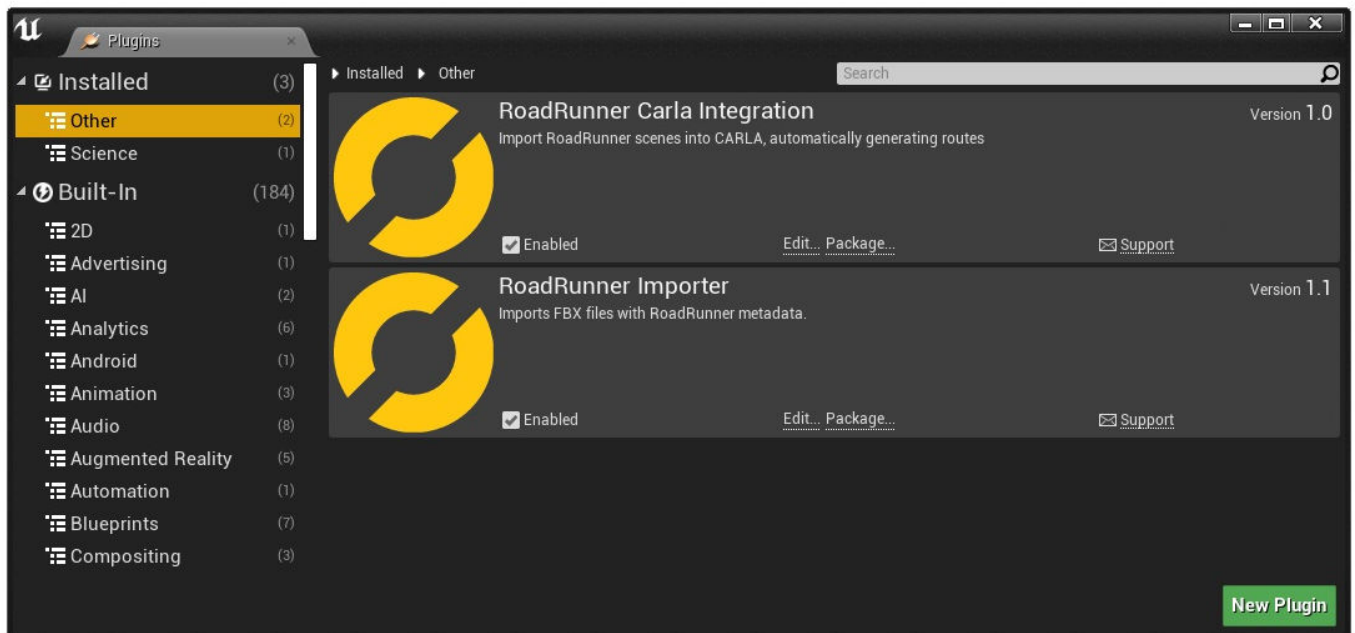
```
$UE4_ROOT/GenerateProjectFiles.sh -project="<Path to Carla folder>/Unreal/CarlaUE4/CarlaUE4.uproject" -game
```

Set UE4_ROOT to your Unreal Engine install directory.

b Open the project and build the plugins by clicking **Yes**.



5 The plugin shows up under **Edit > Plugins**. If it does not appear in that menu, check that the **Enabled** check box is on.



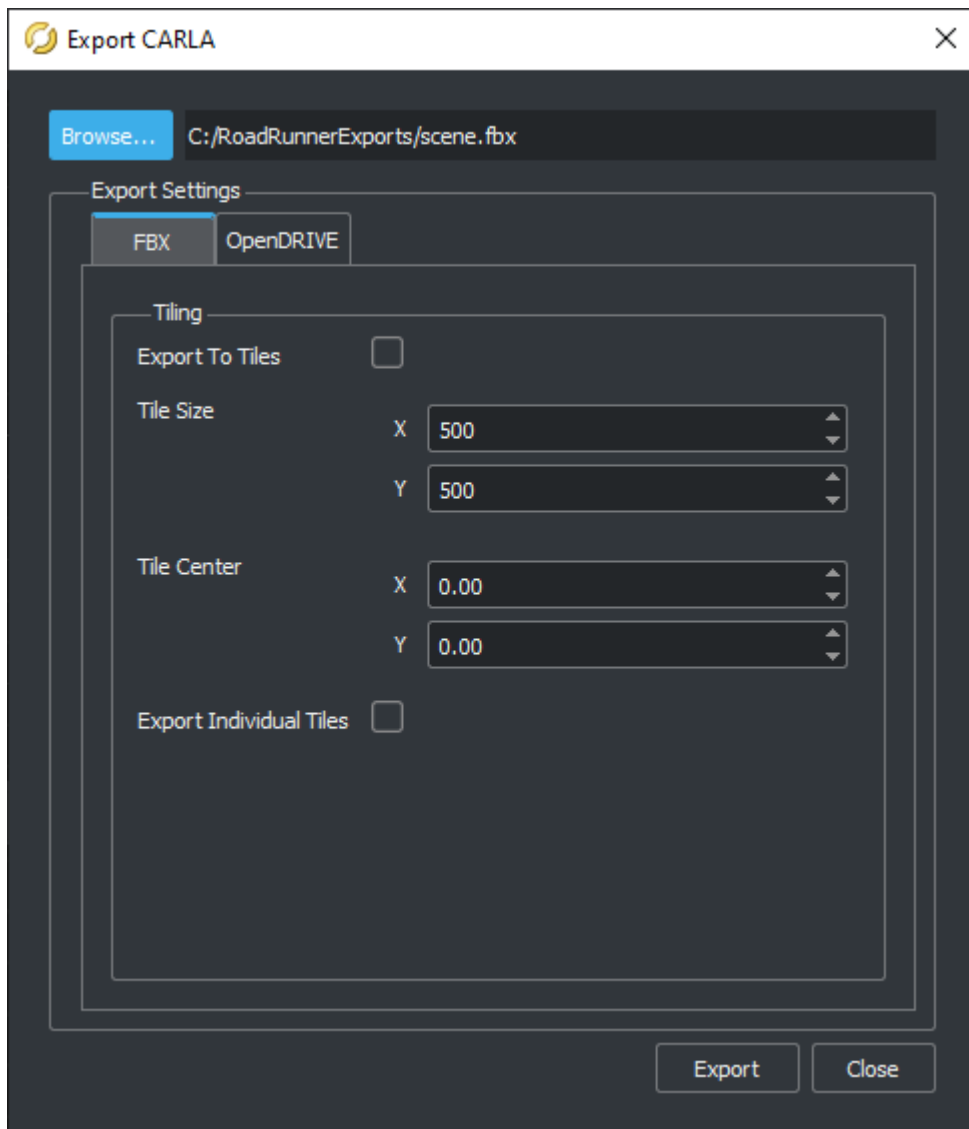
Plugin Contents

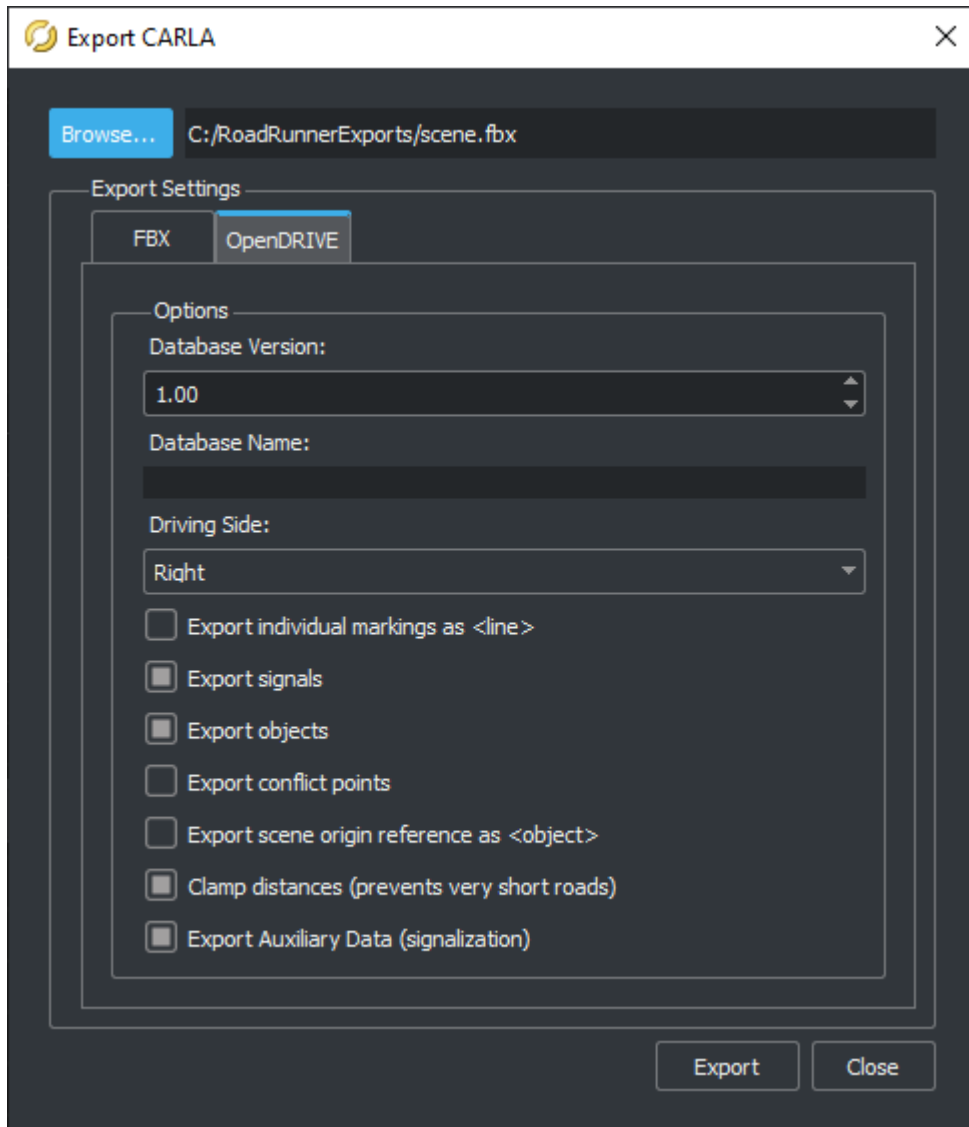
- RoadRunnerImporter module:
 - Overrides the default FBX importer when the metadata file is present
 - Option to overwrite default materials with new materials using the metadata file
 - Import signal data and timing
- RoadRunnerRuntime module:
 - Contains component to control traffic signal visuals
- RoadRunnerCarlaIntegration module:
 - Creates a new map and imports the FBX into the level
 - Moves static mesh assets based on segmentation type
 - Creates materials instantiated from CARLA materials for weather effects
 - Generates the routes from the OpenDRIVE file
- RoadRunnerMaterials plugin:
 - Base materials to create instances from

Exporting from RoadRunner to CARLA

Follow the steps below to export a scene from RoadRunner to Unreal:

- 1 Open your scene in RoadRunner.
- 2 Export the scene using the CARLA option. Select **File > Export > CARLA (.fbx + .xml + .xodr)** from the menu bar.
- 3 In the Export CARLA dialog box, set the mesh tiling on the **FBX** tab and the OpenDRIVE options on the **OpenDRIVE** tab as needed. Then, click **Export**.





- 4 Browse to open the file dialog box to set the exported file's name and path. The FBX, textures, XML, and OpenDRIVE files are exported to the same folder.
- The mesh can be split by segmentation type. Meshes have "<segmentation type>Node" appended to their names.
 - If the **Export To Tiles** option is selected, meshes are split per tile and props are grouped by the tile they are in.
 - By default, only one file is exported. Tiles are stored in separate nodes.
 - If **Export Individual Tiles** is enabled, each tile is stored in its own FBX file.

Note The plugin does not fully support the **Export Individual Tiles** option.

Importing into CARLA/Unreal

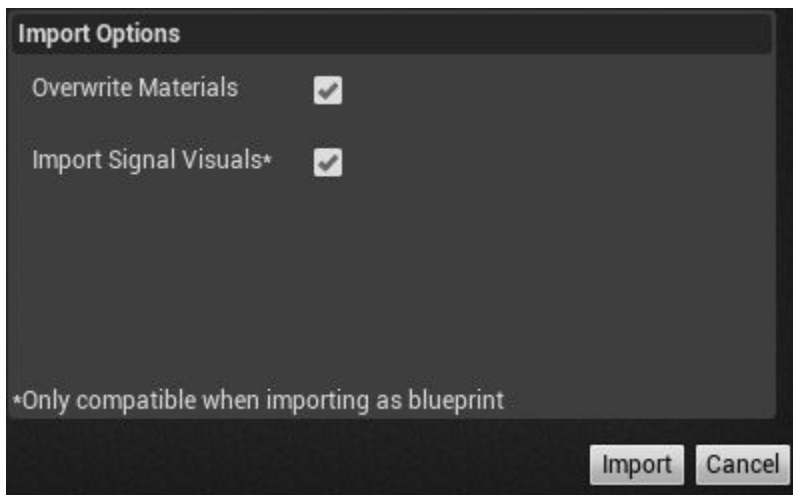
There are multiple ways to import the scene into Unreal for use with CARLA.

- Drag the file into the Content Browser.
- Use the "Import" button and select the FBX file.

The plugin checks if there is a RoadRunner XML file associated with the imported file, and imports as normal if a corresponding XML file is not found.

Selecting **File > Import Into Level** does not use the exported RoadRunner XML and uses the Unreal importer instead.

When the RoadRunner Import Options Dialog Box Opens



- Overwrite Materials
 - Overrides the default material importing. Instances from CARLA materials for roads and foliage.
 - Needs to be unchecked if you want to set the materials to "Use Existing" in the next dialog box.
- Import Signal Visuals
 - Only functional when the "Create one Blueprint asset" option is selected in the next dialog box.

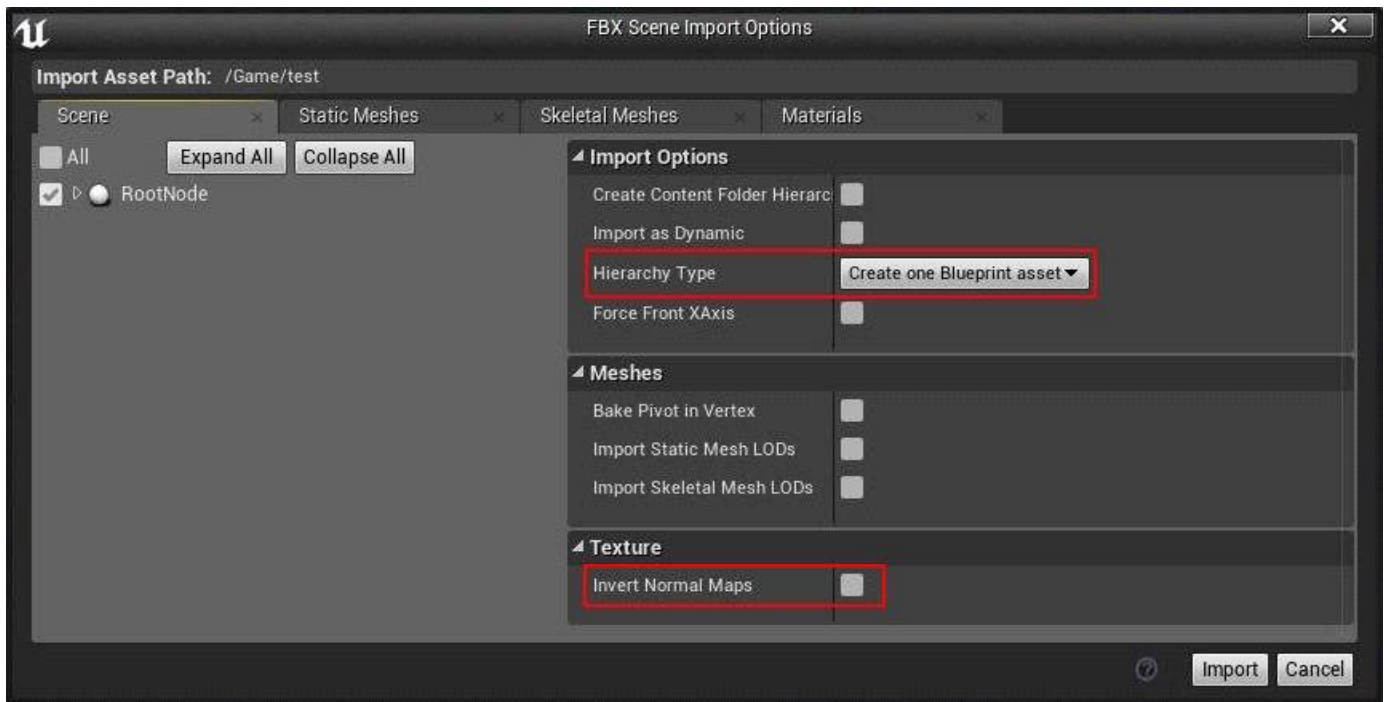
Note The Importing Signal Visuals option does not have any effect on the traffic simulation.

When the FBX Scene Import Options Dialog Box Opens

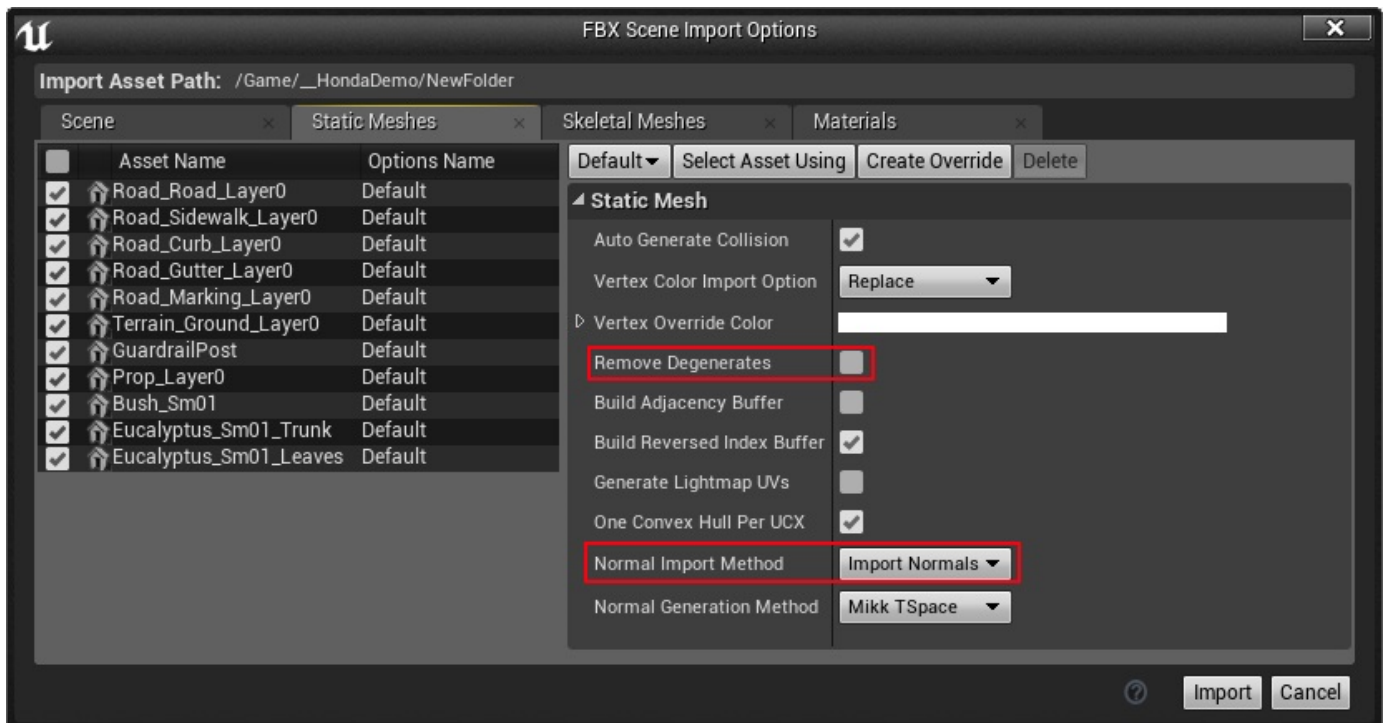
- 1 Set **Scene > Hierarchy Type** to "Create One Blueprint Asset" (selected by default).

Note Only the "Create one Blueprint asset" import option works with materials, signals, and transparency sorting. The "Create one Actor with Components" and "Create Level Actors" options import only materials.

- 2 Select **Invert Normal Maps**, if needed.



- 3 Set **Static Meshes > Normal Import Method** to "Import Normals".



- 4 (Optional) Uncheck **Remove Degenerates**, which can help for some props created in a larger scale.

- 5 Click **Import**.

About Importing Traffic Signals into Unreal

If traffic signals were set up in RoadRunner, they are imported into Unreal as RoadRunnerTrafficJunction components. These controllers are automatically created during import and included in the created blueprint.

The RoadRunnerTrafficJunction component handles the logic for switching between signal states. UUIDs are used to match to specific game objects in the scene.

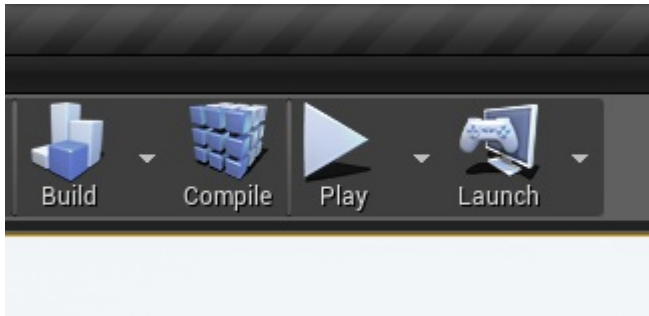
FBX Details

The FBX file will automatically split the mesh by segmentation and transparency sorting layer due to the following:

- Segmentation: CARLA determines segmentation by static mesh assets.
- Transparency sorting: Unreal stores the "Translucency Sort Priority" value on the static mesh component.

Testing the map

- 1 Click **Play** in the editor (the first time you click **Play** takes extra time to build the map).



- 2 Run the example Python® scripts.



Creating Map Packages for Distribution (CARLA 0.9.5)

Note Some steps in CARLA have changed since the 0.9.5 release of CARLA, but the same concepts should apply if there are issues.

Exporting the Map

- 1 Import the FBX into UE4 by using the plugin.
- 2 Run this command: `make export-maps ARGS="--file=<map name>"`
- 3 If the map uses RoadRunner materials, you will likely get a missing material error when trying to run the executable. Follow the steps in the next section if this issue occurs.

RoadRunner Material Workaround

Since CARLA scripts do not copy the RoadRunner plugin "Content" folder, they need to be generated and copied manually to the distribution build.

- 1 In UE4, open the packaging settings under **File > Package Project > Packaging Settings**.
 - a Uncheck "Use Pak File".
 - b Add the map you want to package to the "List of maps to include in a packaged build."
- 2 Run this command: `make package`
- 3 Copy the cooked plugin content (located at `<carla>/Dist/0.9.5-dirty/LinuxNoEditor/CarlaUE4/Plugins/RoadRunnerImporter`) to the desired location.

Note This procedure generates only the materials used in the included maps.

Export to VTD

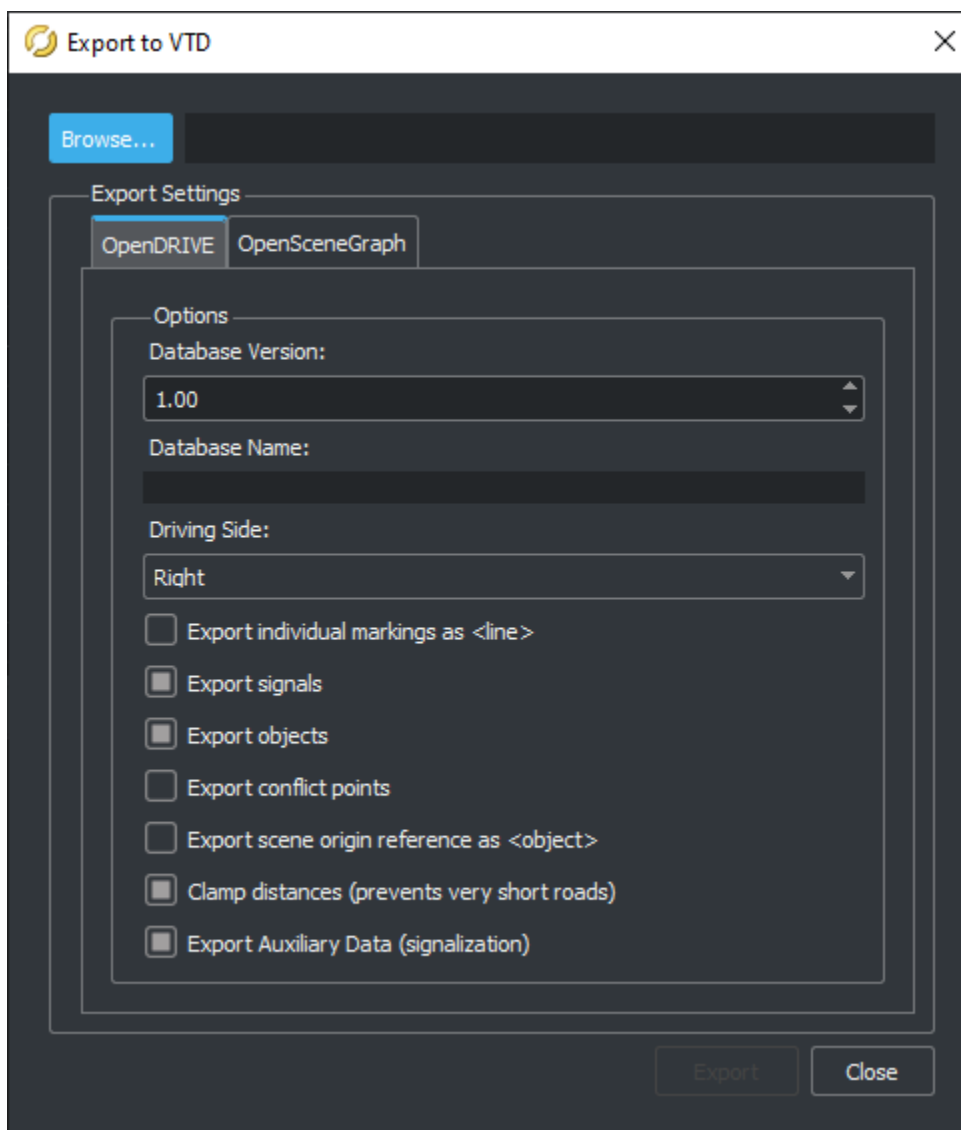
RoadRunner has a combination exporter for exporting scenes to VIRES Virtual Test Drive (VTD)

RoadRunner provides a VTD export option that exports an OpenDRIVE (.xodr) file for the road network and an OpenSceneGraph (.ive) file for the visual scene.

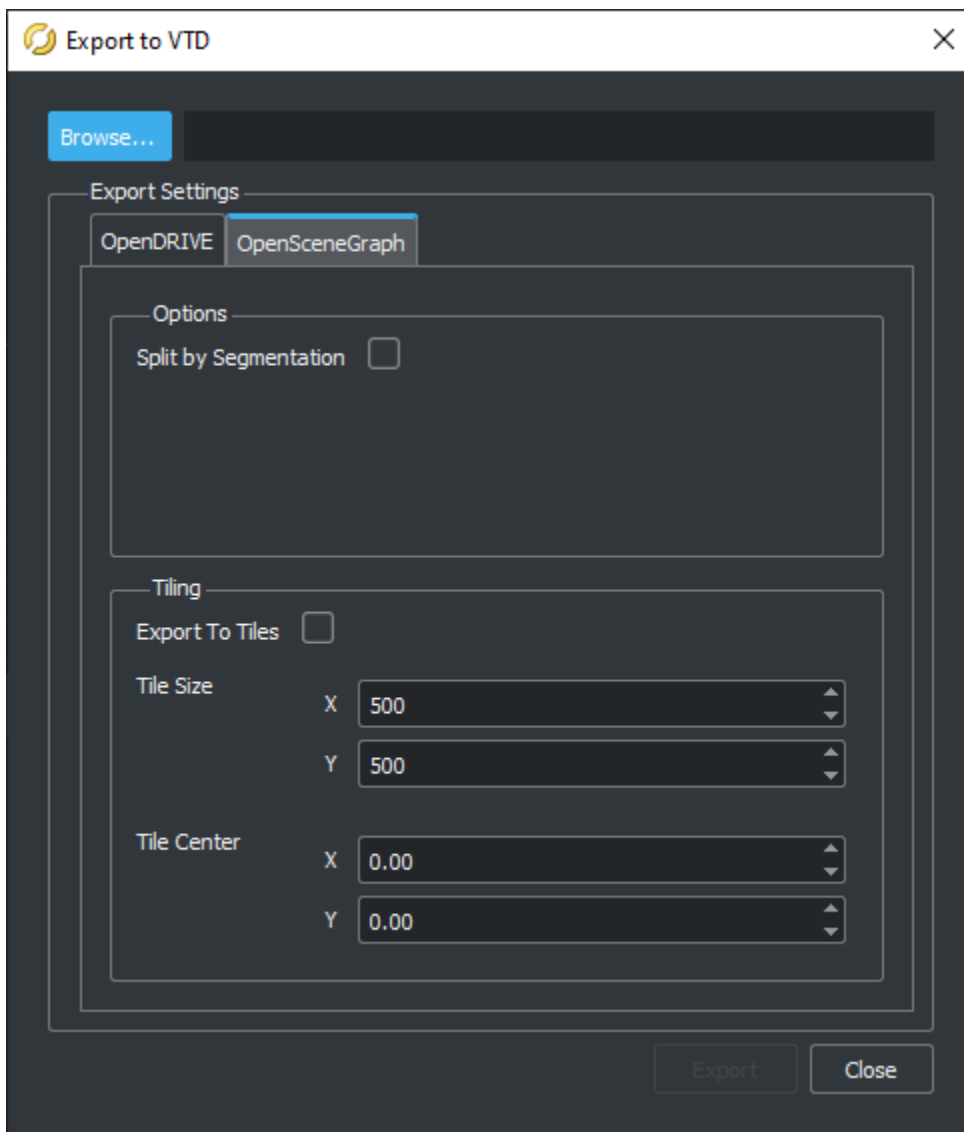
Exporting to VTD

From the menu, select **File > Export > VTD (.xodr + .ive)**.

Export Options (OpenDRIVE)



Export Options (OpenSceneGraph)



Split by Segmentation

This option will split meshes by their segmentation type on page 5-27.

Tiling

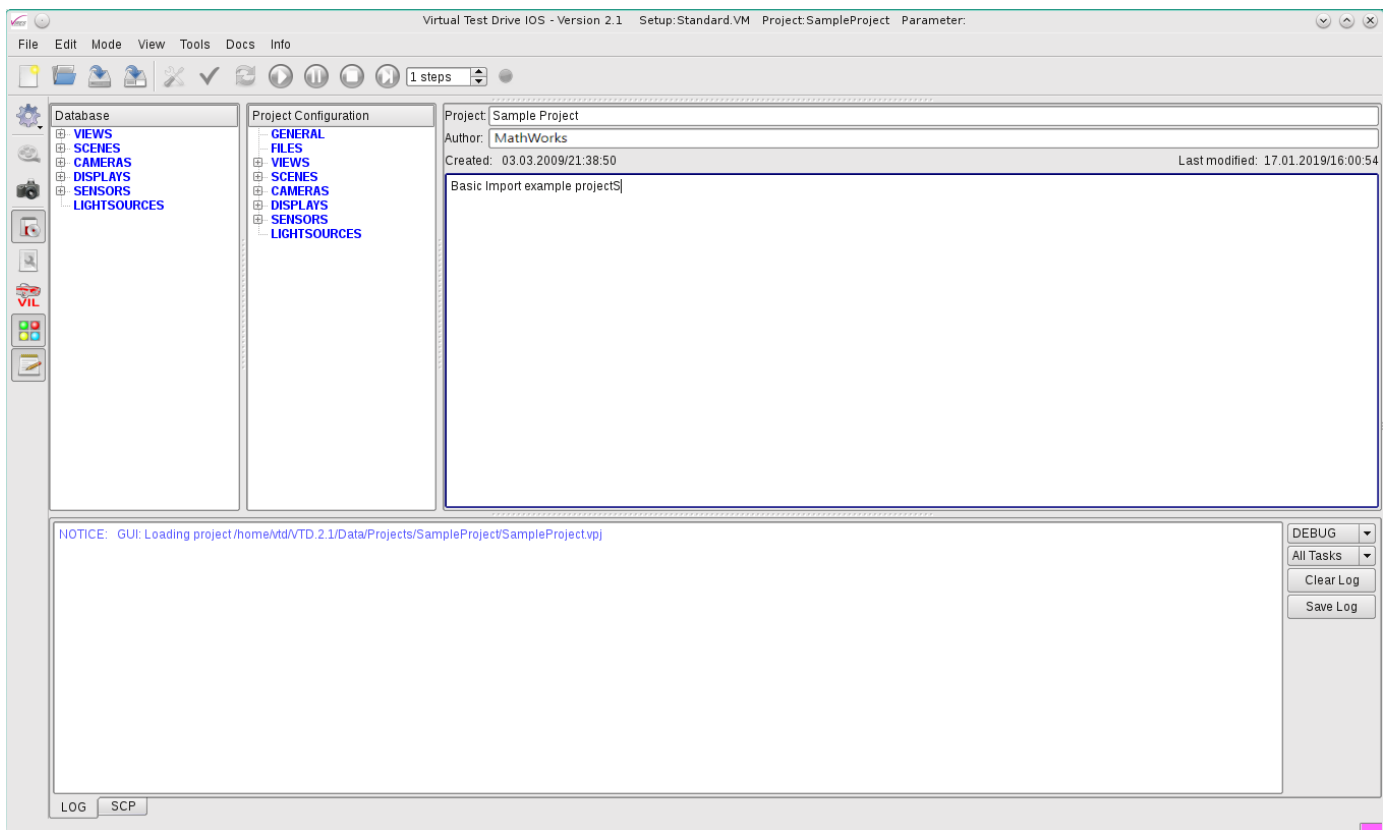
The **Export To Tiles** option splits the meshes per tile. This parameter also groups props by the tile they are in.

- By default, only one file is exported. Tiles are stored in separate nodes.
- VTD does not support scenes tiled to separate files.

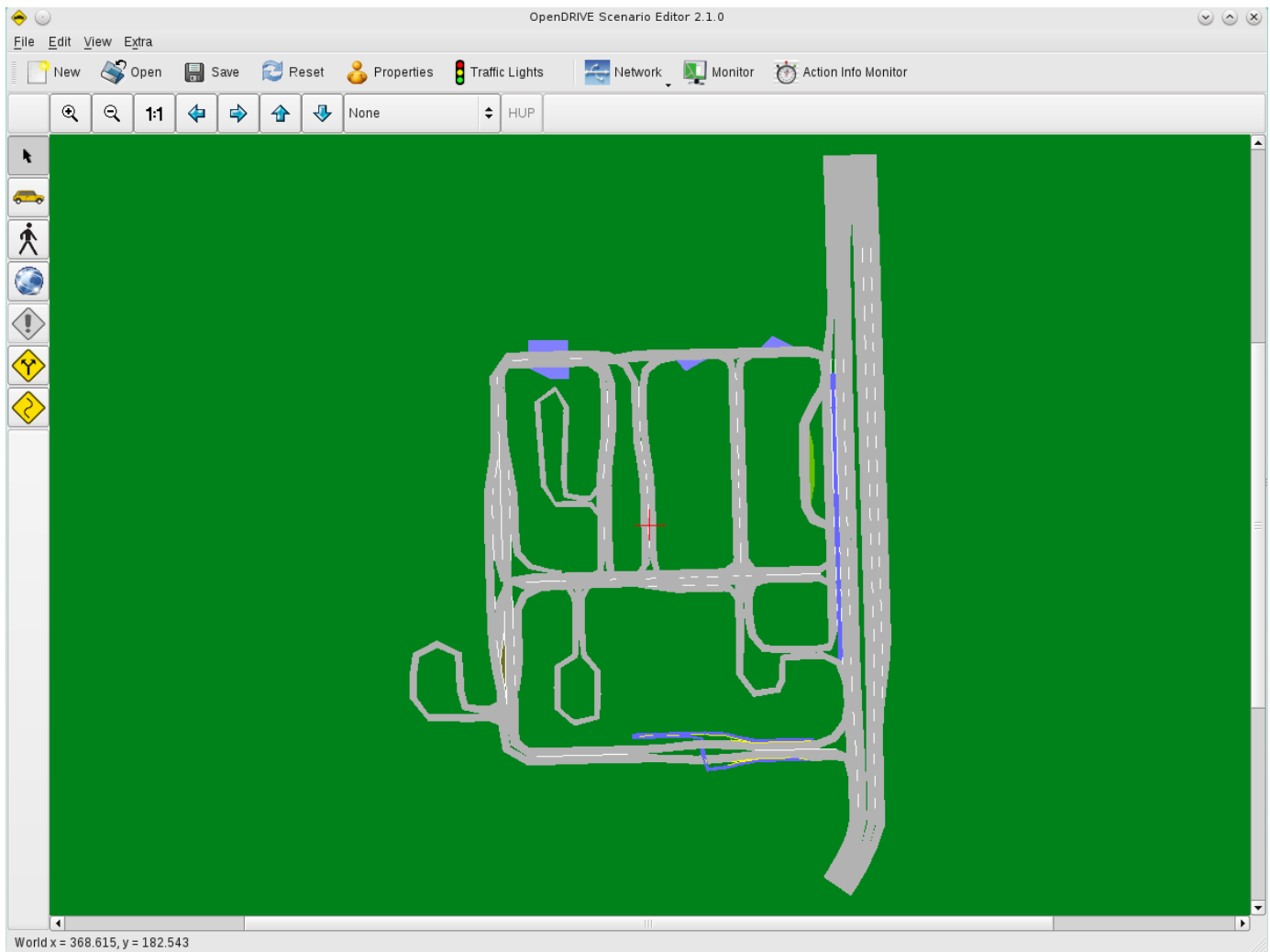
Import into VTD

Once exported from RoadRunner, the scene can be imported into VTD.

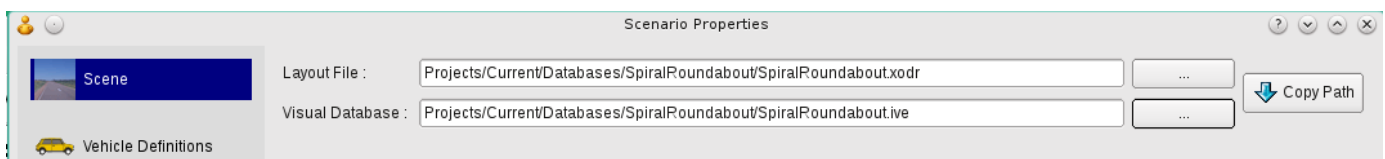
- 1 (Optional) Convert the OpenSceneGraph IVE file to an OSGB file by using OpenSceneGraph version 3.2.3.
- 2 Copy the exported files (OpenDRIVE and IVE files) to your current project in the Databases folder. Placing the files in a separate folder is recommended.
- 3 Open VTD.



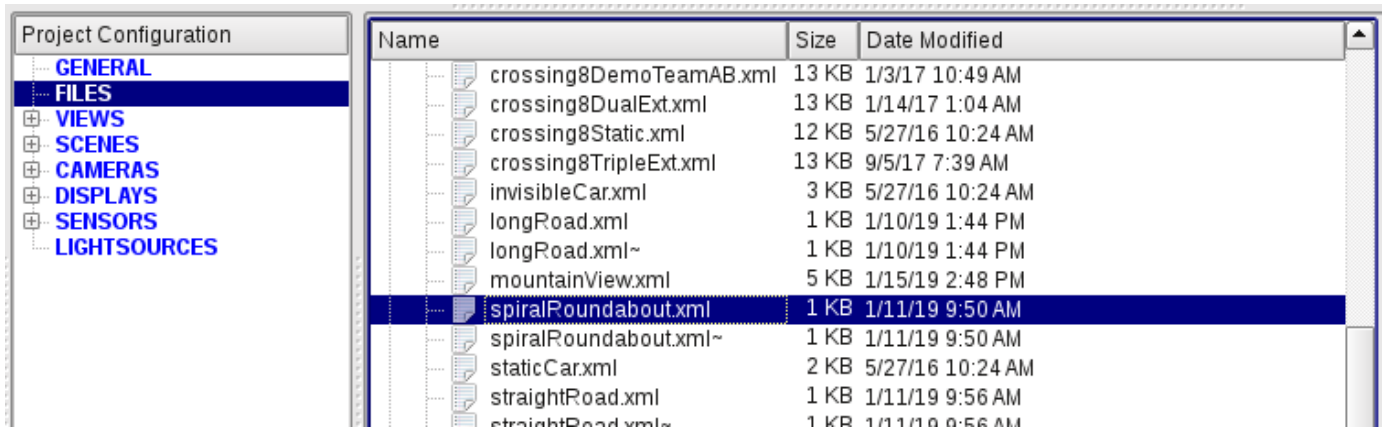
- 4 Open the OpenDRIVE Scenario Editor and click **New**.



- 5 Click the **Properties** button.
- 6 Select the location of the **Layout File** (OpenDRIVE) and **Visual Database** (IVE).



- 7 Insert at least an ego vehicle.
- 8 Save the scenario.
- 9 Select the scenario in VTD.



10 Run the scenario.

Limitations

Refer to “Export to OpenSceneGraph” on page 5-7 for further details on limitations.

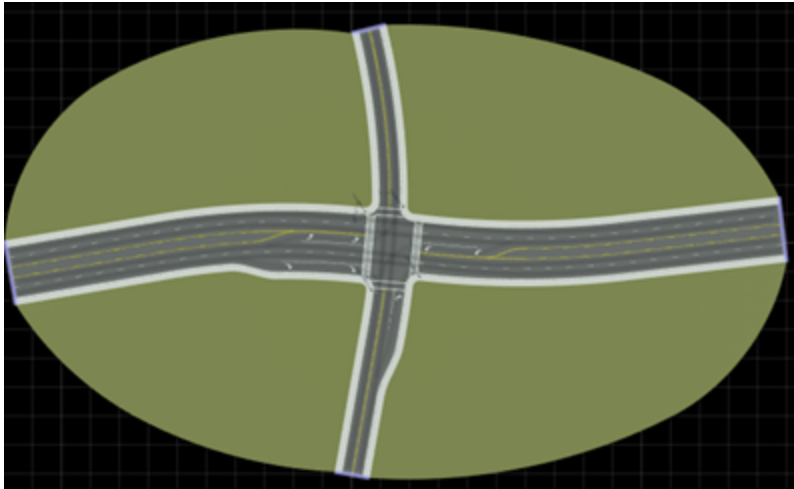
Customize Levels of Detail in Exported Scenes

To improve simulation performance in exported scenes, you can set different levels of detail (LODs) to render at specified distances away from the simulator camera. For example, you can specify lower LODs at distances thousands of meters away, where the reduction in scene quality is acceptable. Export of LODs is supported for FBX files only.

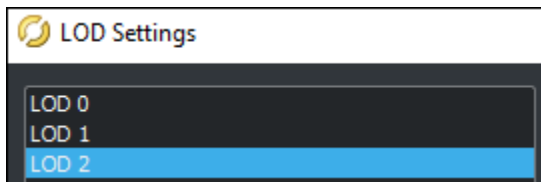
Each LOD you add linearly increases the time it takes to export a scene. However, the substantial performance gains you can achieve by specifying differing LODs can make up for the increased export cost.

Set Levels of Detail in Scene

Open a scene that contains multiple LODs. From the **File** menu, select **Open Scene**. Then, select `FourWaySignal.rsscene`, which is one of the prebuilt scenes included with new RoadRunner projects.



Open the LOD settings for the scene. From the **Tools** menu, select **LOD Settings**. In the left pane of the LOD Settings dialog box, the scene has three LODs: LOD 0, LOD 1, and LOD 2.



You can add new LODs by clicking **Add LOD**. Each LOD you add clones the settings from the previous LOD. RoadRunner does not limit the number of LODs you can add, but each LOD increases export time. The size of the export also increases, because RoadRunner exports a different rendering of the scene at each LOD.

Select the different LODs and notice the change in LOD settings in the right pane. The **LOD Distance** parameter controls the LOD settings to use at distances relative to the simulator camera. This table describes the **LOD Distance** settings for the `FourWaySignal` scene.

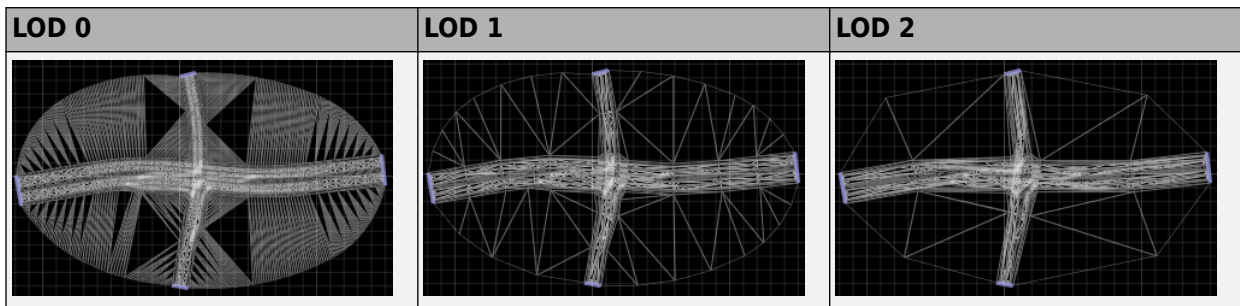
| Level of Detail | LOD Distance | Description |
|-----------------|--------------|--|
| LOD 0 | 1000.0 | Use the defined LOD settings at ranges from 0 to 1000 meters away from the camera. |
| LOD 1 | 2000.0 | Use the defined LOD settings at ranges (1000,2000] meters away from the camera. |
| LOD 2 | 3000.0 | Use the defined LOD settings at ranges 3000 or more meters away from the camera. |

The other dialog box parameters control scene triangulation, overhead scene rendering, and prop packing.

Modify Triangulation Settings

The LOD Settings dialog box includes several parameters that control the number of triangles used to render the scene geometry. Increasing these values reduces the amount of scene triangulation, which speeds up rendering performance at the expense of scene quality.

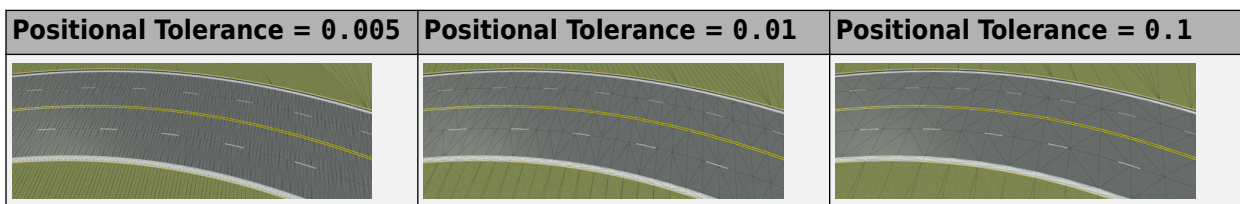
In the `FourWaySignal` scene, view how the triangulation decreases in the scene at the different LOD distances. First, press **F3** to display the shaded wireframe of the scene on the scene editing canvas. Then, in the LOD Settings dialog box, select **Preview Live** and change between LODs in the left pane. The amount of triangulation decreases significantly at each successive LOD.



Press **F3** again to turn off the shaded wireframe.

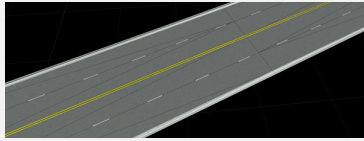
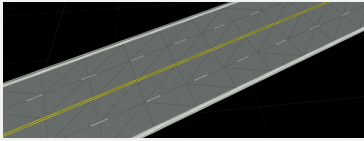
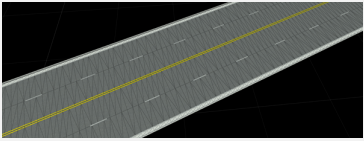
Positional Tolerance Parameter

The **Positional Tolerance** parameter controls the maximum deviation in meters between sampled polylines and their underlying analytical representation. As this value decreases, the number of samples and triangles produced increases. This table shows the difference in triangulation at varying **Positional Tolerance** values.



Max Length Parameter

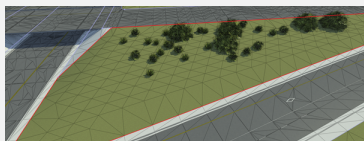
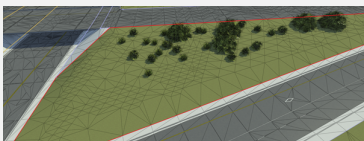
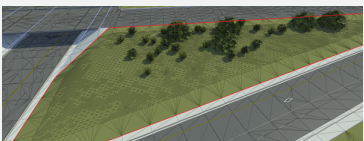
The **Max Length** parameter controls the maximum length between any two samples in the curve triangulation. This table shows the difference in triangulation at varying **Max Length** values.

| Max Length = 50.0 | Max Length = 10.0 | Max Length = 1.0 |
|---|--|---|
|  |  |  |

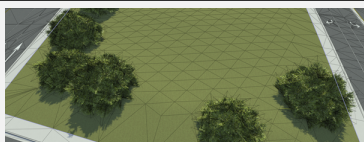
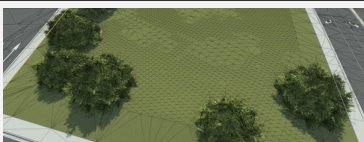
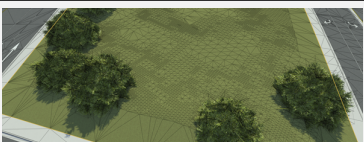
Height Field Sampling Parameters

The LOD Settings dialog box contains several parameters that affect the amount of triangulation used to render elevation changes in the scene.

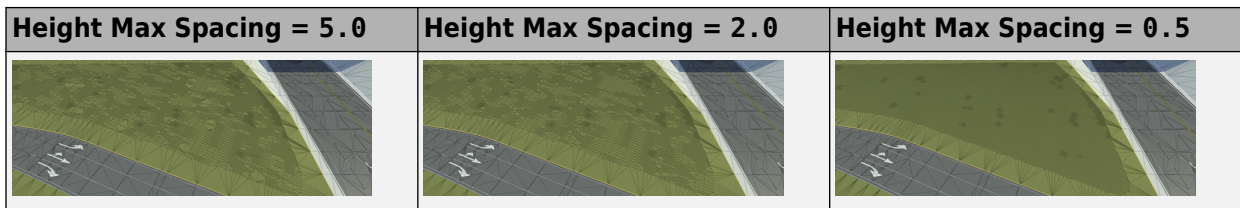
The **Height Tolerance** parameter controls whether to include elevation samples in terrain triangulation relative to the elevation of neighboring samples. Roads are not included in the terrain triangulation. Height tolerance is in meters. As the tolerance decreases, the difference between samples and the underlying elevation field with respect to elevation decreases. Typically, a low tolerance produces a large number of triangles. This table shows the difference in triangulation at varying **Height Tolerance** values.

| Height Tolerance = 0.5 | Height Tolerance = 0.05 | Height Tolerance = 0.005 |
|---|--|---|
|  |  |  |

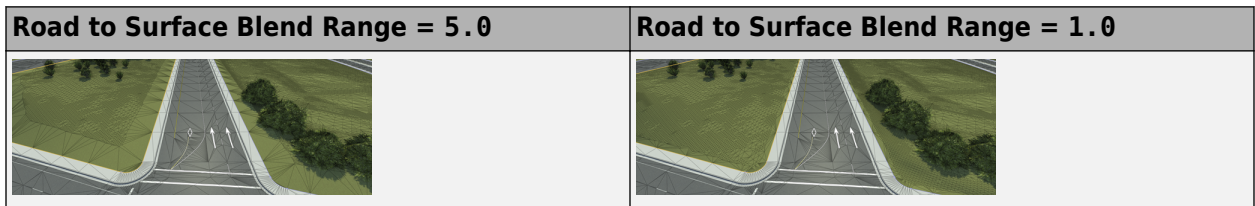
The **Height Min Spacing** parameter controls the minimum distance, in meters, between samples and their neighbors. In general, the lower the tolerance, the closer the samples are allowed to be. Typically, a lower tolerance produces a greater number of triangles. This table shows the difference in triangulation at varying **Height Min Spacing** values.

| Height Min Spacing = 5.0 | Height Min Spacing = 1.0 | Height Min Spacing = 0.1 |
|---|--|---|
|  |  |  |

The **Height Max Spacing** parameter controls the maximum distance, in meters, between samples and their neighbors. In general, the lower the tolerance, the closer the samples are required to be. Typically, a lower tolerance produces a greater number of triangles. This table shows the difference in triangulation at varying **Height Max Spacing** values.



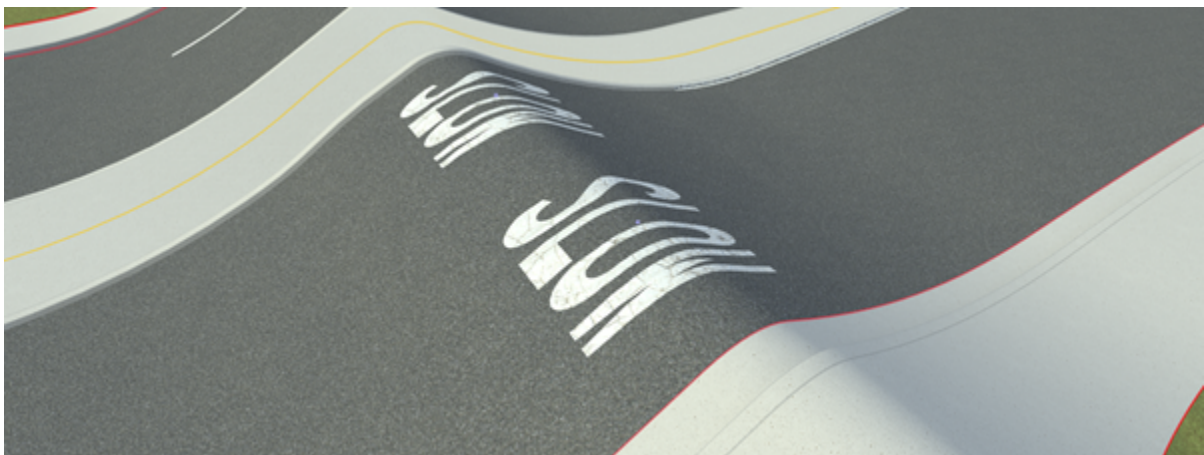
The **Road to Surface Blend Range** parameter controls the amount of distance, in meters, between the surface triangles and any adjacent edge. Typically, a lower blend range produces a greater number of triangles. This table shows the difference in triangulation at varying **Road to Surface Blend Range** values.




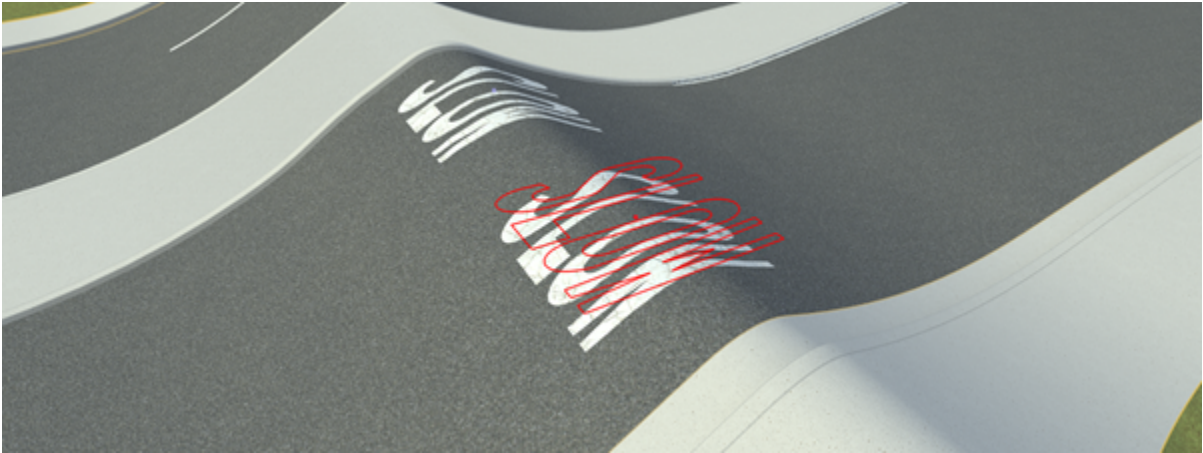
Marking Projection Parameters

Every marking in RoadRunner is projected onto the underlying road or terrain surface to improve visual quality and avoid rendering issues such as z-fighting. The images in this table show the result of projecting markings onto a complex road surface.

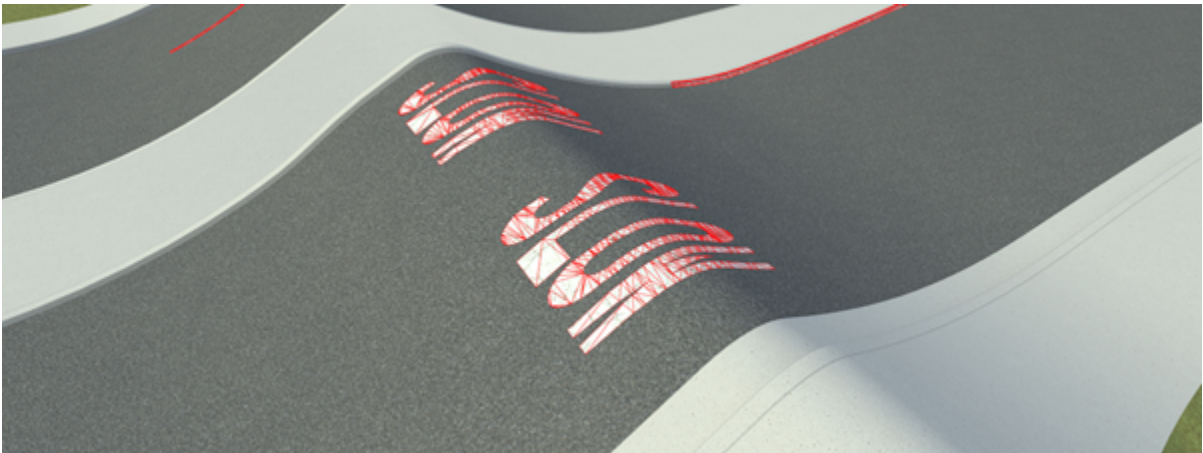
Consider "SLOW" road marking stencils projected onto a road that has an exaggerated height bump.



When you select one of the stencils by using the **Marking Point Tool** , the original flat stencil does not line up well with the underlying surface.

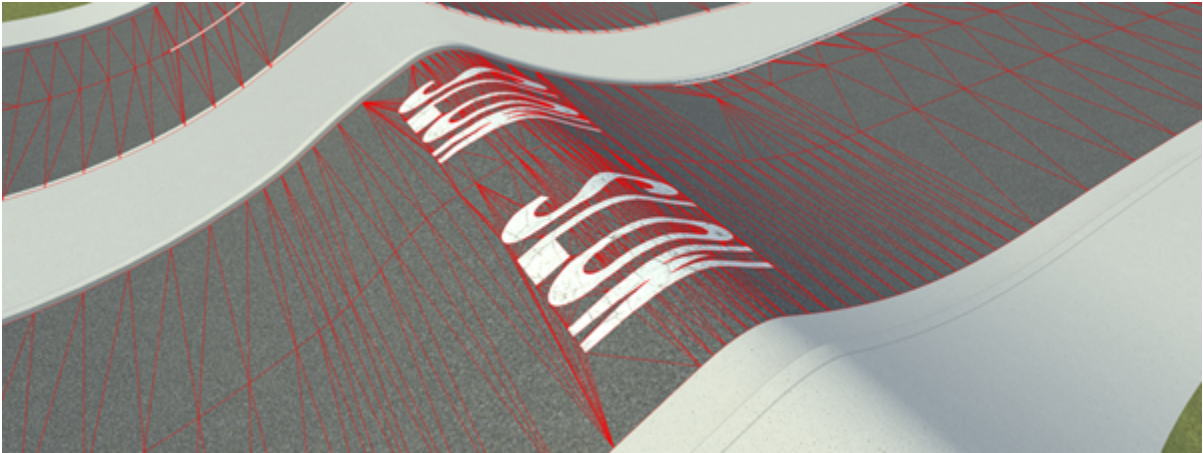


After adjusting the marking projection parameters in the LOD Settings dialog box, the triangulation of the marking outline matches the marking on the road surface.

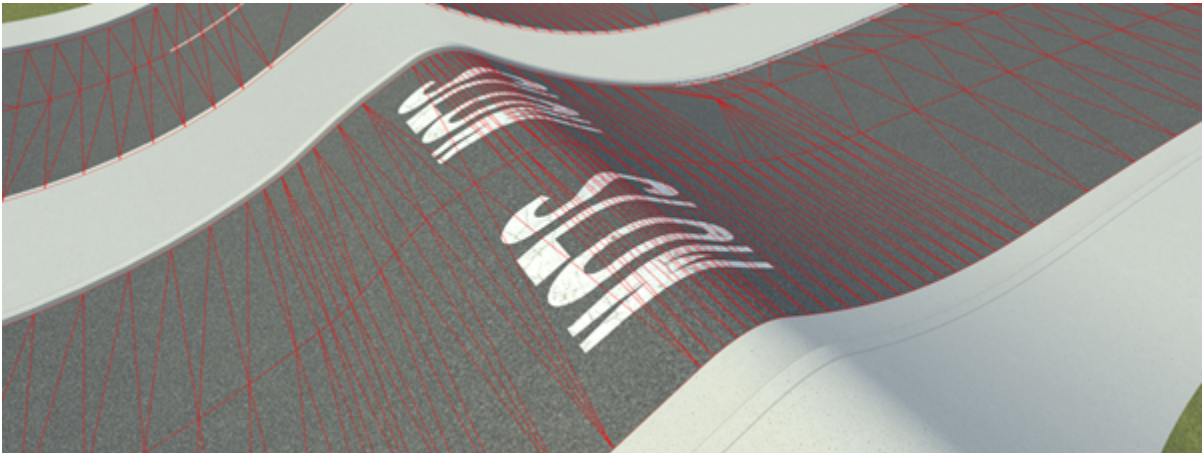


The **Marking Construction Type** parameter controls whether to stitch markings into the underlying road and terrain surface or to float the markings a small distance above the surface once they are projected. Floating markings, also known as decals, produce fewer triangles overall and allow for a more regular road surface triangulation. However, floating markings might require more handling once exported from RoadRunner to avoid z-fighting.


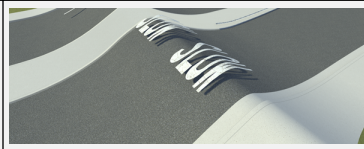
This image shows the sample markings produced when you set **Marking Construction Type** to Cut Out. The asphalt surface is triangulated around the "SLOW" stencil markings.



This image shows the sample markings produced when you set **Marking Construction Type** to **Floating**. The asphalt surface triangulation passes beneath the marking geometry.



The **Floating Marking Offset** parameter controls how much to float a marking above the underlying surface. The lower the value, the smaller the gap between the markings and the surface. With lower values, the likelihood of rendering artifacts, such as z-fighting, increases.

| Floating Marking Offset = 0.01 | Floating Marking Offset = 0.5 |
|---|--|
|  |  |

Modify Scene Rendering

To further improve scene performance, you can select **Overhead Tile Rendering** to flatten the scene into a single texture image per exported tile. RoadRunner flattens roads, lanes, curbs, and terrain into this image but excludes props. Selecting this option can dramatically improve simulation performance. However, the reduction in scene quality is high, so reserve this optimization for high LOD distances only.

In the LOD Settings dialog box for the `FourWaySignal` scene, select `LOD 2`. This LOD has **Overhead Tile Rendering** enabled. You can select additional options to customize the tile rendering:

- **Texture Resolution** — Set the pixel resolution of the textures. Decrease this value to speed up simulation performance at the expense of texture image quality.
- **Use Simplified Mesh** — Simplify the mesh geometry of the texture image into a grid, which reduces the amount of triangles used to render the tile.
- **Simplified Grid Resolution** — Set the number of squares in the triangulation grid. Reduce this value to reduce the number of triangles used to construct the grid at the expense of texture image quality. To enable this option, you must select **Use Simplified Mesh**.

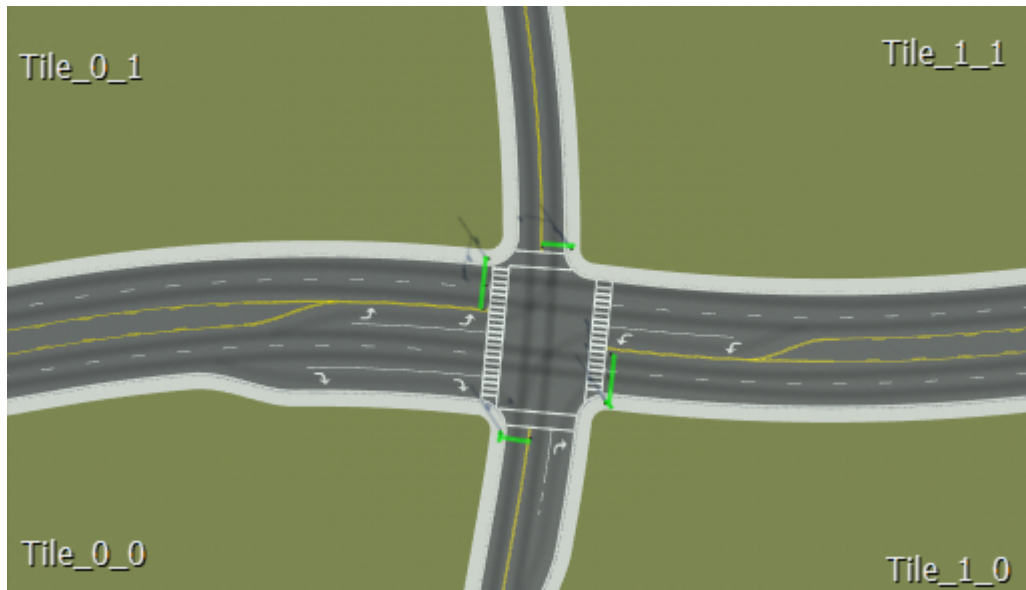
Select **Use Simplified Mesh** and leave **Simplified Grid Resolution** set to 10 to render the tiles using 10-by-10 mesh grids. Click **OK**.

Preview how the texture image looks by clicking the **Scene Export Preview Tool**.

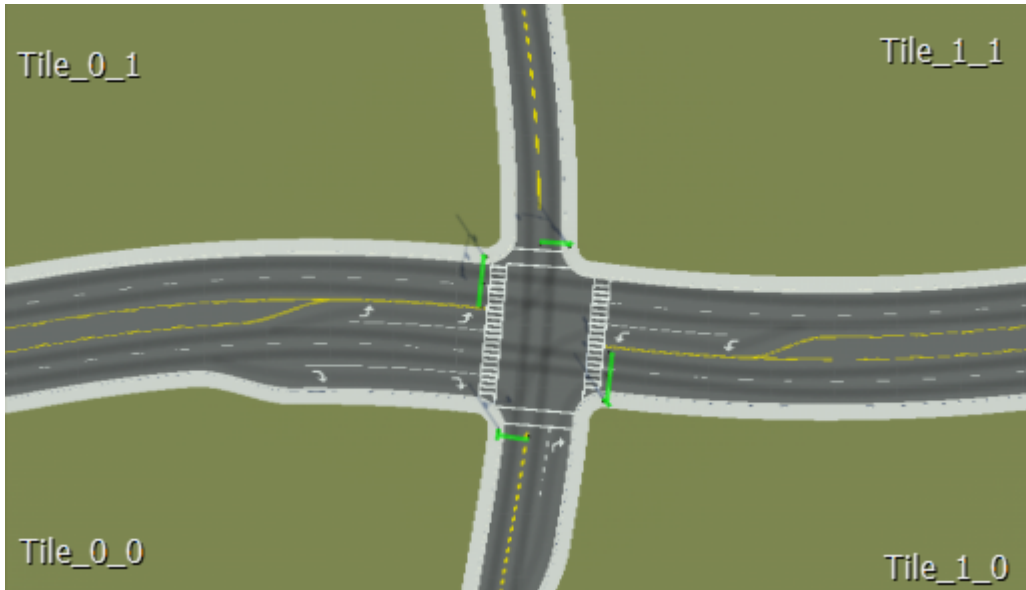
1



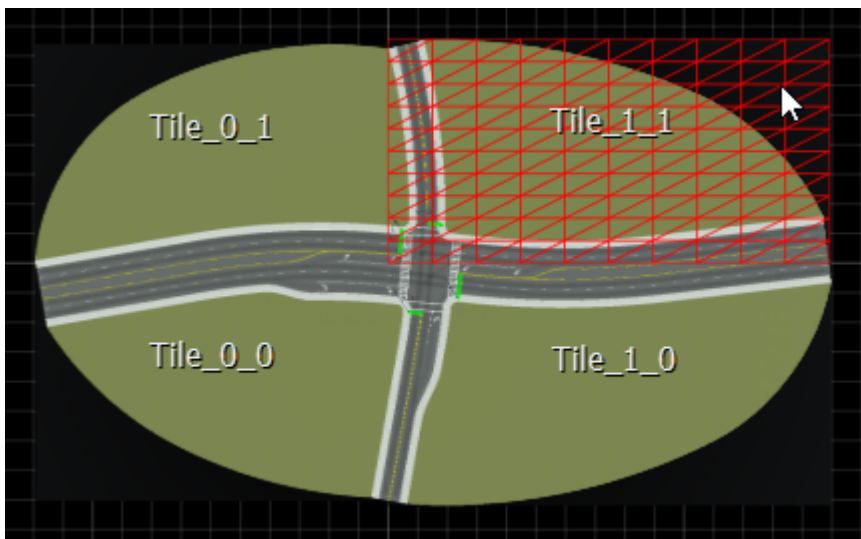
Click the **Scene Export Preview Tool** button. The scene editing canvas displays the four tiles that are exported from the scene.



- 2 In the **Attributes** pane, select **Visualize Camera LOD Distance**.
- 3 Set the **Camera LOD Distance** to 3000, which is the distance at which `LOD 2` starts.
- 4 Click **Refresh Scene**. RoadRunner combines scene objects into one texture image per tile. Only the traffic signal props at the intersection remain selectable. The scene quality is noticeably diminished but in simulators that use this scene, this image is used only when the scene is viewed from 3000 meters away.



- 5 Select the black corner of one of the tiles to view the mesh grid of the tile. The tile is rendered using a simple 10-by-10 triangular mesh.



Pack Props

To improve simulation performance even more, you can pack all props into a single prop model per export tile. With packed props, all textures in a tile are combined into a maximum of two atlas textures: one for opaque materials and one for transparent materials. This optimization can greatly reduce the amount of state changes that the rendering engine of the target simulator uses. Prop packing reduces the amount of texture detail for props and lowers visual quality, so reserve this optimization for high LOD distances only.

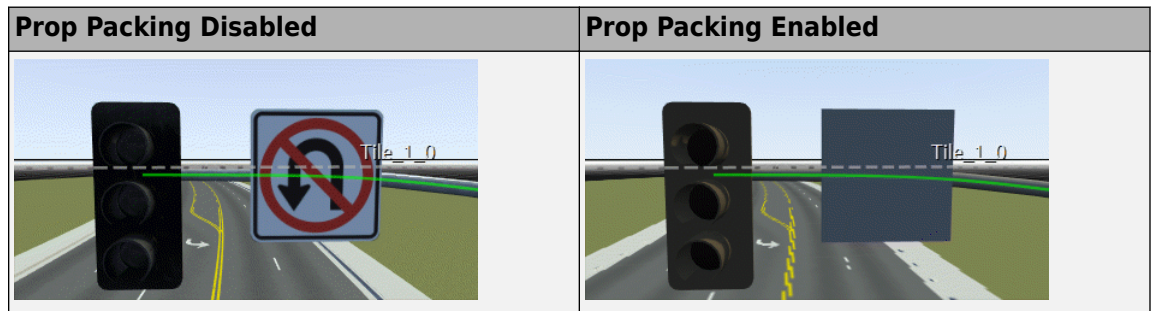
View the effects of prop packing on the FourWaySignal scene.

- 1 Open the LOD Settings dialog box for the FourWaySignal scene. From the **Tools** menu, select **LOD Settings**.

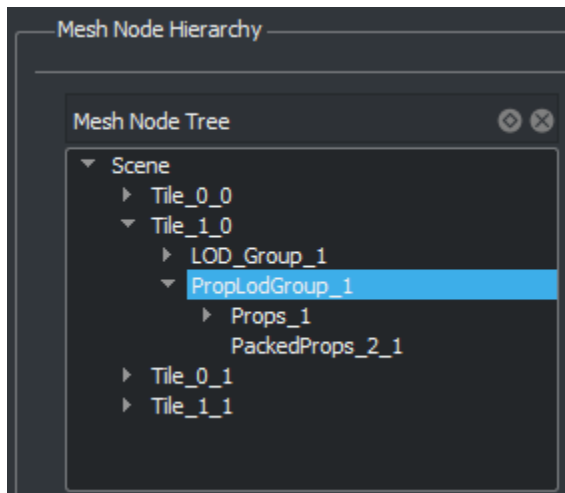
- 2 In the left pane of the dialog box, select LOD 2.
- 3 Select **Pack Props**.
- 4 (Optional) Modify the **Atlas Resolution** parameter. This option changes the number of pixels used to render props. Decrease this value to improve simulation performance at the cost of visual quality.
- 5 Click **OK**.

- 6  Click the **Scene Export Preview Tool** button

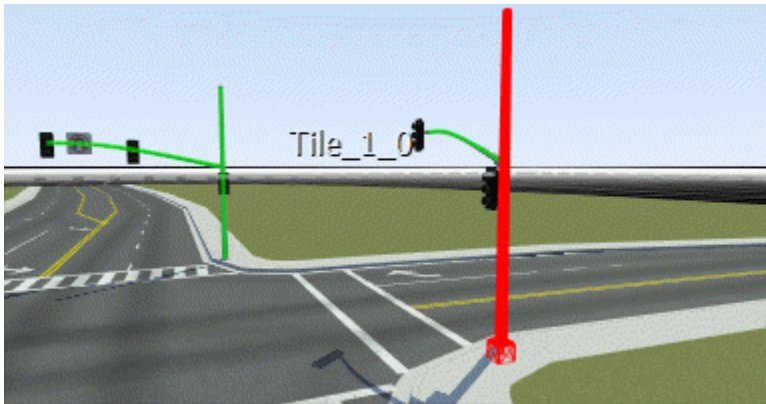
- 7 Focus the camera on the prop assembly in Tile_1_0. This assembly contains a "No U-Turn" sign and traffic lights. In the **Attributes** pane, check that **Visualize Camera LOD Distance** is selected and toggle the **Camera LOD Distance** attribute above and below the LOD 2 threshold of 3000 meters. When the camera distance is at LOD 2, prop packing is enabled and the textures of the prop assembly combine into one model. RoadRunner removes the sign face and simplifies the traffic light textures.



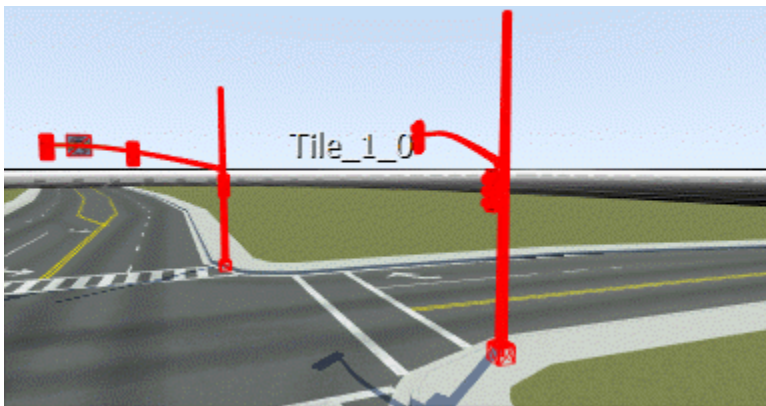
To view all the packed props of a tile, you can select them from the hierarchy of mesh nodes in each tile. In the **Attributes** pane of the **Scene Export Preview Tool**, under **Mesh Node Tree**, navigate to Tile_1_0. Then, select PropLodGroup_1.



The Props_1 group lists every individual prop on the tile, with each prop having its own mesh. Expand this group, select a prop, and view the corresponding mesh selected in the scene editing canvas. For example, this image shows a selected traffic light post prop.



Select the `PackedProps_2_1` mesh node. In the scene editing canvas, all meshes are now collapsed into a single selectable mesh.

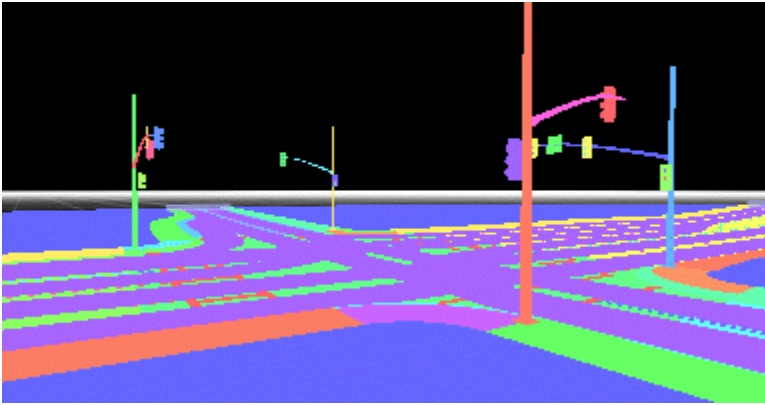


Visualize Performance Improvements

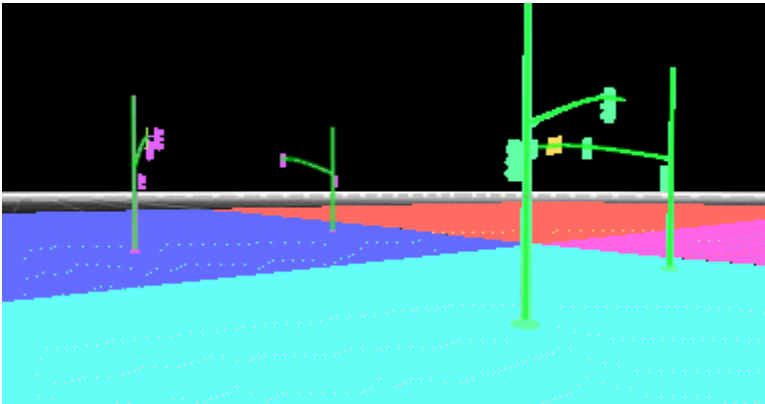
As an additional form of verification, you can visualize the number of draw calls that an external simulator needs to make to render the scene at varying LODs.

From the **View** menu, select **Sensor**, then **Draw Calls**. With this sensor selected, RoadRunner segments the scene by color, where each color represents a separate draw call.

In the `FourWaySignal` scene, with the **Scene Export Preview Tool** still enabled, set the **Camera LOD Distance** attribute below 3000 meters. At these close distances, the simulators make a high number of draw calls to render the roads, terrain, and props.



Set the **Camera LOD Distance** attribute above 3000 meters. At this distance, prop packing and tile rendering are enabled, so the simulators make significantly fewer draw calls to render the scene.



Export Scene

RoadRunner supports the exporting of multiple LODs to FBX files only. If you export to a format other than FBX, then RoadRunner uses the LOD settings specified by LOD 0.

Export the scene as an FBX file. From the **Export** menu, select **Filmbox (.fbx)**.

Open the folder to which you exported the scene. By default, scenes export to the Exports folder of the current RoadRunner project.

The exported FBX data includes the texture atlases used at the varying levels of detail. For more details on the exported FBX data, see “Export to FBX” on page 5-3.

See Also

Scene Export Preview Tool

Related Examples

- “Export to FBX” on page 5-3

Export Custom Formats

You can export RoadRunner scenes to a variety of file formats. These exporting options are available from the **File** menu, under **Export**. You can also configure this menu to include custom export options that combine the different file formats that RoadRunner supports. To specify a custom configuration, follow these steps:

- 1 Create an XML file that configures the details of the custom export format.
- 2 Save the XML file to the **Export** folder of the RoadRunner project that you want to contain the export option and name it `ExportConfigurations.xml`.

Create Export Configuration XML File

To enable custom exports, you must specify an export configuration XML file similar to the one shown here. This sample file specifies a custom option that exports an FBX file, OpenDRIVE file, GeoJSON file, and RoadRunner scene metadata file together in one zip file.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<ExportConfigurations>
  <Configuration name="My Custom Format" zip="true" extension="zip">
    <Format name="Filmbbox" extension="fbx">
      <Option name="ExportToTiles" value="true" type="bool"/>
      <Option name="ExportIndividualTiles" value="true" type="bool"/>
      <Option name="TileSize" value="1000,1000" type="vec2"/>
      <Option name="Center" value="0,0" type="vec2"/>
      <Option name="SpecularMapAsRoughnessMap" value="true" type="bool"/>
      <Option name="Offset" value="0,0,-72.226989" type="vec3"/>
    </Format>
    <Format name="OpenDRIVE" extension="xodr">
      <Option name="ExportIndividualRoadMarkings" value="false" type="bool"/>
      <Option name="ExportSignals" value="true" type="bool"/>
      <Option name="ExportObjects" value="true" type="bool"/>
      <Option name="ExportConflictPoints" value="false" type="bool"/>
      <Option name="ExportSceneReference" value="false" type="bool"/>
      <Option name="QuantizeRoads" value="true" type="bool"/>
      <Option name="VerticalOffset" value="-72.226989" type="double"/>
    </Format>
    <Format name="GeoJSON" extension="geojson"/>
    <Format name="Scene Metadata" extension="xml"/>
  </Configuration>
</ExportConfigurations>
```

<ExportConfigurations> Element

The top-level element of the XML file, `<ExportConfigurations>`, is required. This element contains all the custom export configurations that appear in the **File** menu, under **Export**.

<Configuration> Elements

The `<ExportConfigurations>` element can contain one or more `<Configuration>` elements. Each `<Configuration>` element corresponds to a custom export format that appears in the **File** menu, under **Export**. This table describes the attributes that you can specify for this element.

| <Configuration> Attribute | Optional or Required | Description |
|---------------------------|----------------------|--|
| name | Required | Name of the exported file, specified as a string. |
| extension | Required | Extension of the exported file, specified as a string. |
| zip | Optional | Option to export the custom format as a zipped file, specified as <code>false</code> or <code>true</code> . If you specify <code>zip</code> as <code>true</code> , then specify <code>extension</code> as a zip file extension, such as <code>zip</code> or <code>rar</code> . Default: <code>false</code> |

<Format> Elements

Each <Configuration> element can contain one or more <Format> elements. These elements correspond to the file formats that RoadRunner exports as part of the custom configuration. This element requires a `name` attribute, which specifies the name of a file format that RoadRunner supports. This element also includes an optional `extension` attribute, which specifies the extension of the exported file. If you do not specify this attribute, then the exported file has the default extension for that file format.

This table lists the format names that you can specify for this attribute and their corresponding valid extensions. These attributes are case sensitive. If you specify an invalid attribute, then the export dialog box does not include a tab for that format.

| name Attribute | Valid extension Attributes | Description | Export Format Details |
|----------------|-----------------------------|---|--|
| AutoCAD | <code>dxf</code> (default) | Export scene to an AutoCAD file. | "Export to AutoCAD" on page 5-2 |
| Filmbox | <code>fbx</code> (default) | Export scene to a Filmbox file. | "Export to FBX" on page 5-3 |
| glTF | <code>glTF</code> (default) | Export scene to a GL Transmission Format (glTF) file. | "Export to glTF" on page 5-5 |
| OpenFlight | <code>flt</code> (default) | Export scene to an OpenFlight file. | "Export to OpenFlight" on page 5-6 |
| OpenSceneGraph | <code>osg</code> (default) | Export scene to an OpenSceneGraph file. | "Export to OpenSceneGraph" on page 5-7 |
| USD | <code>usd</code> (default) | Export scene to a Universal Scene Description (USD) file. | "Export to USD" on page 5-14 |
| Wavefront | <code>obj</code> (default) | Export scene to a Wavefront OBJ file. | "Export to Wavefront OBJ" on page 5-8 |
| OpenDRIVE | <code>xodr</code> (default) | Export scene to an OpenDRIVE file. | "Export to OpenDRIVE" on page 5-15 |

| name Attribute | Valid extension Attributes | Description | Export Format Details |
|----------------|----------------------------|---|---|
| Apollo | xml (default) | Export scene to the Baidu Apollo file format. | “Export to Apollo” on page 5-32 |
| GeoJSON | geojson (default) | Export scene to a GeoJSON file. | “Export to GeoJSON” on page 5-9 |
| Scene Metadata | xml (default) | Export scene metadata to an XML file. | “RoadRunner Metadata Export” on page 5-30 |

<Option> Elements

Each <Format> element can optionally include <Option> elements. These elements specify options for configuring your export formats. If you do not specify an option, then RoadRunner assigns the default value to that option. Each <Option> element that you do specify must include these attributes:

- name — Name of option
- value — Value of option
- type — Data type of option

This table describes the option attributes that you can specify for mesh export formats, which apply to these formats:

- AutoCAD
- Filmbox
- glTF
- OpenSceneGraph
- USD
- Wavefront

| name Attribute | Description | value Attribute | type Attribute |
|---------------------|--|--|----------------|
| SplitBySegmentation | Split meshes by their segmentation type. For more details, see “Segmentation” on page 5-27. | false (default) true | bool |
| PowerOfTwoTextures | Resize the dimensions of exported textures by rounding them up to the next highest power of two. | false (default) true | bool |
| ExportToTiles | Split meshes per tile. | false (default) true | bool |
| TileSize | Specify the size of exported tiles, when ExportToTiles is true. | 0,0 (default) two-element real-valued vector | vec2 |

| name Attribute | Description | value Attribute | type Attribute |
|---------------------------|---|--|-----------------------|
| ExportIndividualTiles | Export tiles to separate files. | false (default) true | bool |
| EmbedTextures | Embed the exported textures inside the exported file. | false (default) true | bool |
| LodAsPercentage | Specify levels of detail by percentage. | false (default) true | bool |
| CollapseMesh | Export the scene as a single mesh rather than as multiple nodes. | false (default) true | bool |
| SplitByPass | Split meshes by triangulation pass, for example, separating markings or damage from the road surface. | false (default) true | bool |
| SplitByMaterial | Split meshes by material. | false (default) true | bool |
| SpecularMapAsRoughnessMap | Use the specular map as the roughness map. | false (default) true | bool |
| UnityLods | Export levels of detail for use with Unity. | false (default) true | bool |
| Offset | Specify additional (x,y,z) offset for exported scene. | 0,0,0 (default) three-element real-valued vector | vec3 |

This table describes the option attributes that you can specify for mesh export formats, which apply to these formats:

- OpenDRIVE
- Apollo

| name Attribute | Description | value Attribute | type Attribute |
|-----------------------|---|----------------------------|-----------------------|
| QuantizeRoads | Perform quantization to prevent very short roads or lanes. | true (default) false | bool |
| DatabaseVersion | Specify an identifier for the exported scene, which is useful for versioning exports of the same scene. | 1.0 positive real scalar | double |
| DatabaseName | Specify the name of the exported scene. | "" string | string |

| name Attribute | Description | value Attribute | type Attribute |
|------------------------------|--|-----------------------------|----------------|
| ExportIndividualRoadMarkings | Export road markings using the <line> definition in OpenDRIVE. | false (default) true | bool |
| ExportSignals | Export all signals and signs mapped to junctions as <signal> entries. | true (default) false | bool |
| ExportObjects | Export all props as <object> entries. | true (default) false | bool |
| ExportConflictPoints | Export conflict points, that is, point in junctions at which connecting roads overlap. | false (default) true | bool |
| ExportSceneReference | Include <object> element at (0,0) in exported file to use as a reference point. | false (default) true | bool |
| ReduceFileSize | Remove new lines from exported file to reduce file size. | false (default) true | bool |
| VerticalOffset | Specify vertical offset for exported scene. | 0 nonnegative real scalar | double |

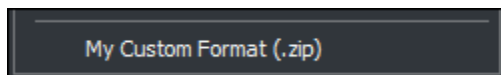
For the GeoJSON format, ReduceFileSize is the only valid option.

The Scene Metadata format does not have any options.

Save Export Configuration File to Project

Save the XML file containing your export configurations to the **Project** folder of your project and give it the name `ExportConfiguration.xml`. If you do not save the file to this folder and with this name, then RoadRunner does not recognize the custom export formats.

After saving the file, restart RoadRunner and open a scene in the project that has the `ExportConfiguration.xml` file. In the **File** menu, under **Export**, the custom format options are now included at the bottom of the list. This figure shows a sample custom format option.



See Also

Related Examples

- "Export Scenes"

- “RoadRunner Project and Scene System” on page 2-2

RoadRunner Asset Library Product Overview

RoadRunner Asset Library Product Description

Populate RoadRunner scenes with a library of 3D models

RoadRunner Asset Library is a set of 3D models and assets for 3D scenes created with RoadRunner. The library provides hundreds of models, including road and highway signs, traffic signals, road surface markings, trees, barriers, and road damage textures, such as cracks and oil spills. All models are professionally designed and visually consistent.