MAK TECHNOLOGIES

VR-Engage Users Guide



Version 2.0.1

ST Engineering North America

Copyright © 2022 MAK Technologies
All rights Reserved. Printed in the United States.

Under copyright laws, no part of this document may be copied or reproduced in any form without prior written consent of MAK Technologies.

VR-Exchange™, VR-TheWorld™, VR-Vantage™, DI-Guy™, and DI-Guy Scenario™ are trademarks of MAK Technologies. MÄK Technologies®, VR-Forces®, RTIspy®, B-HAVE®, and VR-Link® are registered trademarks of MAK Technologies.

GL Studio® is a registered trademark of The DiSTI® Corporation.

Portions of this software utilize SpeedTree® technology (©2008 Interactive Data Visualization, Inc.). SpeedTree is a registered trademark of Interactive Data Visualization, Inc. All rights reserved.

SilverLining™ is a trademark of Sundog Software.

Terrain Profiles are based in part on the work of the Qwt project (http://qwt.sourceforge.net).

3ds Max® is a registered trademark of Autodesk, Inc.

All other trademarks are owned by their respective companies.

For third-party license information, see "Third Party Licenses" on page 19 in the Preface.

MAK Technologies
10 Fawcett Street, Suite 204
Cambridge, MA 02138 USA

Voice: +1 617.876.8085
Fax: +1 617.876.9208

info@mak.com

www.mak.com

Revision VRN-2.0.1-1-220826



Contents

Preface

How the Manual Is Organized 14
Documentation Set 15
MAK Products 15
How to Contact Us 18
Telephone 18
E-mail 18
Internet 18
Post 18
Document Conventions 18
Mouse Button Naming Conventions 19
Third Party Licenses 19
Boost License 22
libXML and libICONV 23

1. Introduction to VR-Engage

1.1. VR-Engage Features 26
1.1.1 VR-Engage Versions 26
1.1.2 Simulation (Role-Playing) Capabilities 26
1.1.3 User Interface 31
1.1.4 Image Generator 31
1.1.5 Terrain 32
1.2. Integration with VR-Forces 32
1.3. Integration with Other MAK Products 33
1.3.1 After Action Review – Using MAK Data Logger 33
1.3.2 Network Interoperability with Other Federates 33
1.4. Extensibility 33
1.5. Licensing 33

1.5.1 Vortex License (for Ground Vehicle Dynamics) 34
1.5.2 RTDynamics License (For Aircraft) 34
1.5.3 License Requirements for Hosting a VR-Forces Session 34

2. Installing VR-Engage

2.1. Installing VR-Engage 38
2.1.1 Installing VR-Engage on Windows 38
2.1.2 Uninstalling VR-Engage from Windows 39
2.1.3 Installing the makData Package 39
2.2. Upgrading VR-Engage Simulation Model Sets 40
2.3. The VR-Engage Directory Structure 40
2.4. Installing and Setting Up the MAK License Manager 41
2.4.1 Specifying the License Server 41
2.5. Installing an RTI 43
2.5.1 Installing the MAK RTI 44
2.6. Input Devices 45
2.6.1 HMD Support 45
2.7. Factors that Affect Performance 45
2.8. Localizing the Graphical User Interface 46

3. Starting VR-Engage

3.1. Starting VR-Engage 49
3.1.1 Hosting a VR-Forces Session 49
3.1.2 Starting an Unhosted Session 50
3.1.3 Admin, User, and Lockdown Modes 51
3.2. Choosing a Role 52
3.2.1 Choosing an Existing Role 53
3.2.2 Creating a New Entity 54
3.2.3 Using an HMD with an Entity 55
3.2.4 Assigning Roles from VR-Forces 56
3.3. Connecting to VR-Forces 58
3.4. Changing Your Role 59
3.4.1 Changing Your Role in the Choose Role Panel 59
3.4.2 Changing Your Role by Embarking on a Vehicle 59
3.5. Entity Engagement and Scenario State 60
3.6. Overview of the Application Menu 61
3.7. Starting a New Session 62
3.8. Configuring the Simulation Connection 62
3.9. Playing in Windowed Mode 63
3.10. Introduction to Head-Mounted Displays 64
3.10.1 Prerequisites for Using an HMD 64
3.10.2 Resetting the Tracking Sensor 64

3.10.3 Calibrating the Eye Tracking for Varjo ... 65
3.10.4 Enabling and Disabling Virtual Reality Rendering ... 65
3.10.5 Configuration Options for Virtual and Mixed Reality ... 66
3.10.6 Setting Up Mixed Reality ... 67
3.10.7 Optimizing Performance with HMDs ... 70
3.11. Creating a Startup Configuration ... 70
3.11.1 Minion and Boss Configuration ... 71
3.12. Starting Multiple Simulation Engines ... 72
3.13. Command-Line Options ... 72
3.14. Command-Line Options for Player Stations ... 85
3.15. Managing Plug-ins ... 85
3.15.1 Loading Plug-ins ... 86
3.15.2 Adding a Plug-in ... 86
3.15.3 Specifying the DLLs for a Plug-in ... 87
3.15.4 Adding a Plug-in Configuration ... 88
3.15.5 Deleting a Plug-in Configuration ... 89
3.15.6 Deleting a Plug-in ... 89

4. Common Player and Entity Features

4.1. The Action Menu ... 92
4.1.1 Selecting Options on the Action Menu ... 92
4.2. Using the Radio ... 93
4.2.1 Using MAK Radio in VR-Forces ... 94
4.2.2 Using the Standalone Radio Application ... 94
4.2.3 Using the Intercom or a VOX Radio ... 96
4.2.4 Configuring the MAK Radio ... 96
4.2.5 Audio Settings ... 100
4.2.6 Jamming Radios ... 101
4.3. Changing the View Mode ... 101
4.4. Starting VR-Forces Events ... 102
4.5. Introduction to Sound Effects ... 102
4.5.1 Enabling Sound Effects ... 103
4.5.2 Configuring Sound Effects ... 103
4.5.3 Decibels Configuration ... 104
4.5.4 Associating Sound Files with Entities and Events ... 105
4.6. Resetting Entities ... 105
4.7. HMD Controls ... 106
4.8. Saved Observer Views ... 106
4.8.1 Selecting a View ... 107
4.8.2 Transitioning to a View ... 107
4.8.3 Resetting the View ... 107

5. Controlling People

5.1. Introduction 110
5.1.1 Enabling and Disabling Mouse Control 110
5.2. Basic Movements 111
5.2.1 Moving Your Character Around the Terrain 111
5.2.2 Changing Your Character's Posture 111
5.2.3 Looking Around 111
5.2.4 Gesturing and Signaling 111
5.2.5 Climbing Ladders 112
5.3. Navigation 112
5.3.1 Using the Compass 112
5.3.2 Using the Minimap 113
5.4. Health and Stamina 114
5.4.1 Sustaining Damage to Health 115
5.4.2 Experiencing Fatigue and Losing Stamina 115
5.5. Weapons 116
5.5.1 Choosing a Weapon 116
5.5.2 Aiming a Weapon 116
5.5.3 Firing a Weapon 118
5.5.4 Firing Weapons with an HMD Controller 118
5.5.5 Throwing a Hand Grenade 118
5.5.6 Deploying and Stowing Weapons 119
5.5.7 Reloading a Weapon 120
5.6. Javelin Missile Launchers 120
5.6.1 Using the Shoulder-Mounted Missile Launcher 121
5.6.2 Using the Tripod-Mounted Missile Launcher 121
5.7. Using the AN/PED-1 LLDR 123
5.8. Embarking on a Vehicle 124
5.8.1 Embarking on the Stryker's Interior 125
5.8.2 Disembarking from a Vehicle 126
5.9. Carry or Drag a Person 126
5.10. Parachuting 127
5.11. Tasking Other Entities 127
5.11.1 Disembark Entity 128
5.11.2 Follow Me 128
5.11.3 Move to Location 128
5.11.4 Move to My Location 129
5.11.5 Suppress Location 129
5.12. Turning on Your Flashlight 130
5.13. Using the Binoculars 130
5.14. Changing Visual Sensors 131
5.15. Lasing Targets 131

5.15.1 Defining the Laser Code 132
5.15.2 Displaying a Gimbaled Sensor's View 133
5.15.3 Enabling Laser Designators 133
5.16. Opening and Closing Doors and Windows 133

6. Controlling Ground Vehicles

6.1. Introduction 136
6.1.1 Ground Vehicle Roles 136
6.1.2 Ground Vehicle Display Options 137
6.2. Driving Ground Vehicles 137
6.2.1 Starting the Engine 137
6.2.2 Putting the Vehicle into Gear 137
6.2.3 Controlling the Throttle 138
6.2.4 Steering a Vehicle 138
6.2.5 Releasing and Engaging the Parking Brake 138
6.2.6 Locking and Unlocking the Differential (LAV Only) 138
6.2.7 Turning the Headlights On and Off 139
6.3. The Gunner Role 139
6.3.1 Firing a Vehicle's Gun 139
6.3.2 Choosing a Weapon 139
6.3.3 Moving the Turret 140
6.3.4 Stabilizing a Vehicle's Gun 140
6.4. The Commander Roles 140
6.4.1 Overriding the Gunner 141
6.4.2 Slewing the Gun to Designate a Target 141
6.4.3 Using the Laser Designator 141
6.5. Using the Intercom 141
6.6. Changing the Sensor for a Vehicle Crew Member 142
6.7. Pivoting a Tank 142
6.8. Emergency Lights and Sirens 142
6.9. Configuring Ground Vehicles on Geocentric Terrains 142

7. Controlling Aircraft

7.1. Introduction 147
7.1.1 Aircraft Roles 147
7.1.2 The Head-Up Display 147
7.1.3 The Instrument Panel 149
7.1.4 Aircraft Display Options 151
7.2. Taking Control of an Aircraft 152
7.3. Using Autopilot 153
7.3.1 Setting the Autopilot Mode 153
7.4. Selecting a Bullseye 154

7.5. Using the Radar Warning Receiver 154
7.6. Controlling Landing Gear 155
7.7. Controlling Flaps 156
7.8. Launching Countermeasures (Chaff and Flare) 156
7.9. Choosing a Weapon 156
7.10. Air-to-Ground Combat 157
7.10.1 Targeting an Air-to-Ground Weapon Using MGRS Coordinates 157
7.10.2 Targeting an Air-to-Ground Weapon Using a Laser Code 158
7.11. Air-to-Air Combat 158
7.11.1 Selecting Targets 159
7.11.2 Firing an Air-to-Air Weapon 162
7.11.3 Focusing the Sensor on the Radar Target 162
7.12. Using the Common Operating Picture 162
7.12.1 Displaying COP Information for an Entity 163
7.12.2 Toggling the COP Focus Area 163
7.12.3 Selecting an Entity for the COP 164
7.13. Requesting SAR Images from RadarFX Server 164
7.13.1 Requesting a SAR Image 164
7.13.2 Configuring a SAR Request 165

8. Controlling Drones

8.1. Introduction 168
8.1.1 Drone Camera Display 168
8.1.2 Video and Images 169
8.1.3 Drone Battery Life 169
8.2. Operating the Drone 169
8.2.1 Deploying and Flying the Drone 170
8.2.2 Stopping and Resuming Use of the Drone 170
8.2.3 Switching Control Between Character and Drone 170
8.2.4 Landing the Drone 171
8.2.5 Returning the Drone to its Starting Position 171
8.2.6 Recovering the Drone 171

9. Sensor Operator Role

9.1. The Sensor Operator Role 174
9.1.1 Focusing the Sensor 175
9.1.2 Changing the Sensor Mode 176
9.1.3 Resetting the Sensor 176
9.1.4 Video and Pictures 176
9.1.5 Defining a Laser Code 176
9.1.6 Lasing Targets 177
9.1.7 Firing Weapons 177

9.2. Creating a Remote Camera Entity 177
9.3. Placing Sensors on Remote Entities 178
9.3.1 Configuring the Location of the Remote Camera 178
9.4. Configuring the Sensor Operator Overlay 181

10. Video Stream and Image Capture

10.1. Introduction 186
10.2. Video Streams 186
10.2.1 Streaming Video as a Sensor Operator 186
10.2.2 Streaming Video from a Drone 186
10.2.3 Specifying the Video Stream File Settings 187
10.2.4 Viewing a Video File 187
10.2.5 Receiving a Video Stream 187
10.3. Capturing Images 188

11. Mouse, Keypad, and Controller Mappings

11.1. Mouse and Keyboard Mappings 191
11.1.1 Application Management 191
11.1.2 Action Menu Controls 191
11.1.3 Human Controls 192
11.1.4 Ground Vehicle Controls 193
11.1.5 Aircraft Controls 194
11.2. Game Controller Mappings for Using the Action Menu 195
11.3. Game Controller Mappings for Human Control 196
11.4. Game Controller Mappings for Drones 197
11.5. Game Controller Mappings for the LAV and Tanks 197
11.6. Game Controller Mappings for Cars and Trucks 199
11.7. Racing Wheel Controllers 199
11.8. Flight Controllers 201
11.8.1 HOTAS Warthog Mappings for Fixed-Wing Entities 202
11.8.2 HOTAS Warthog Mappings for Rotary-Wing Entities 205
11.8.3 T.Flight Hotas X Mappings for Fixed-Wing Entities 207
11.8.4 T.Flight Hotas X Mappings for Rotary-Wing Entities 210
11.8.5 Game Controller Flight Control Mappings for Fixed-Wing Entities 213
11.8.6 Game Controller Flight Control Mappings for Rotary-Wing Entities 214
11.8.7 Game Controller Mappings for the Sensor Operator 214
11.9. Head-Mounted Display Controllers 215
11.10. Game Controller Button Numbering 217
11.11. Tips for Logitech Gamepad 219
11.12. Configuring Games Controllers and Keyboard Control 219
11.12.1 Windows Game Controllers 220
11.12.2 File Structure 220

11.12.3 Mappings 221
11.12.4 Input Types 223
11.12.5 Adding a New Input Device 225
11.12.6 Player Controls State 227
11.12.7 Component Mapping Groups 229
11.12.8 List of Actions 230

12. Controllable Entities

12.1. VR-Forces Entities that VR-Engage Can Control 240

13. Adding and Editing VR-Engage Entity Models

13.1. The VR-Engage Simulation Model Set 276
13.2. Ground Vehicles and Vortex 276
13.3. Extend the VR-Engage SMS 278
13.3.1 Creating a New Simulation Model Set 278
13.3.2 Creating a New VR-Engage-Controllable Entity 279
13.3.3 Editing a VR-Engage-Controllable Entity 280
13.3.4 Deploying the SMS to VR-Engage 280
13.4. VR-Engage Extensions to .entity Files 281
13.4.1 Structure of Role Data in .entity Files 281
13.4.2 The vrEngageRoles and role Entries 281
13.4.3 The parameterOverride Entry 283
13.4.4 Role-Slot Association 284
13.4.5 Roles Included with VR-Engage 284

14. Model Customization Tutorial

14.1. Overview of Customizing a Model 288
14.2. Extending VR-Engage.sms 288
14.3. Editing the Entity in the SOE 289
14.3.1 Promote the Entity to your SMS 289
14.3.2 Review the Entity's Systems 290
14.4. Editing the Tank's .entity File 290
14.4.1 Specify a VR-Engage-Capable Platform 291
14.4.2 Add the VR-Engage Tag 291
14.4.3 Add the Driving Role Component 291
14.5. Test Driving Your Tank 292
14.5.1 Copy the SMS Files 292
14.5.2 Specify Your SMS as the Default 292
14.5.3 Drive Your Tank 293
14.6. Creating the Views for Your Tank 293
14.6.1 Create a New Driver Views File 294

14.6.2 Override Tank Driver View with Your View 294

15. Creating VR-Engage Scenarios in VR-Forces

15.1. Installing the VR-Forces/VR-Engage Add-ons 298
15.2. Creating a VR-Engage Scenario in VR-Forces 298
15.3. Running Your Scenario 300
15.3.1 Hosting a Scenario in VR-Forces 300
15.3.2 Hosting a Scenario in VR-Engage 300

16. Customizing Overlays

16.1. Overview of Custom Overlays 302
16.1.1 Types of Overlays 302
16.1.2 Examples of Each Type of Overlay 302
16.2. Adding a QML Overlay 303
16.2.1 Testing Your QML Overlay 304
16.3. Adding a Custom Sensor Overlay 305
16.4. Adding a GL Studio Overlay 305
16.4.1 Creating a GL Studio RSO 306
16.4.2 Creating VR-Vantage GL Studio Updaters 306
16.4.3 Configuring the VR-Vantage GL Studio Model Definition 306
16.4.4 Modifying Role Configuration Scripts 307

17. Activating Scripts in VR-Forces

17.1. Activating Lua Scripts in VR-Forces 310
17.2. Adding a Script to the Action Menu 310
17.3. Activating a Script 311

A. Troubleshooting

A.1. Performance and Display Issues 314
A.2. The Fault Tolerant Heap 314
A.3. Issues with VRAM 314
A.4. Remote Camera Placement 315
A.5. Other Limitations 315

B. Embarkation Slot Configuration in VR-Engage

B.1. Overview of Embarkation Slots 318
B.2. Door Point-type Slots for Scripted Doors on Walkable Vehicles 318
B.2.1 Ingress vs. Egress Door Points 318
B.2.2 Example Door Point Slot Configuration 318
B.2.3 Door Point Parameters 319
B.3. Embark Point-type Slots 320

B.3.1 Example Embark Point Slot Configuration 320
B.3.2 Embark Point Parameters 321

Index **323**



Preface

This manual is for persons who will install or use VR-Engage. The manual assumes that you are familiar with basic administrative tasks and the graphical window environment for your operating system.

Please see release documentation for information about changes and updates to VR-Engage since this manual was completed.

How the Manual Is Organized

This manual is organized as follows:

- "Chapter 1. Introduction to VR-Engage" on page 25 describes the basic features of VR-Engage.
- "Chapter 2. Installing VR-Engage" on page 37 explains how to install VR-Engage software, the license manager, and an RTI.
- "Chapter 3. Starting VR-Engage" on page 47 explains how to start and stop VR-Engage, load a terrain, and connect to a simulation.
- "Chapter 4. Common Player and Entity Features" on page 91 describes features that apply to all platform types.
- "Chapter 5. Controlling People" on page 109 explains how to play a human character, including moving, changing weapons, firing weapons, and embarking and disembarking.
- "Chapter 6. Controlling Ground Vehicles" on page 135 explains how to drive ground vehicles, including the LAV, M1A2, and civilian vehicles.
- "Chapter 7. Controlling Aircraft" on page 145 explains how to control aircraft and how to request SAR images.
- "Chapter 8. Controlling Drones" on page 167 explains how to deploy and control the quadcopter drone.
- "Chapter 9. Sensor Operator Role" on page 173 explains the sensor operator role and how to control the sensor interface.
- "Chapter 10. Video Stream and Image Capture" on page 185 describes how to stream video or save images to a file from a quadcopter drone or sensor operator.
- "Chapter 11. Mouse, Keypad, and Controller Mappings" on page 189 illustrates the mouse, keyboard, and game controller mappings configured in VR-Engage.
- "Chapter 12. Controllable Entities" on page 239 lists the entity types that you can control in VR-Forces from VR-Engage if VR-Forces is running a scenario that uses VR-Engage.sms.
- "Chapter 13. Adding and Editing VR-Engage Entity Models" on page 275 explains how to extend the VR-Engage simulation model set.
- "Chapter 14. Model Customization Tutorial" on page 287 provides a simple example of how to create customize a model to make it available for player control.
- "Chapter 15. Creating VR-Engage Scenarios in VR-Forces" on page 297 describes what is involved in creating your own scenarios.
- "Chapter 16. Customizing Overlays" on page 301 describes how to edit entity configurations to customize what players can see and do.
- "Chapter 17. Activating Scripts in VR-Forces" on page 309 explains how to add scripts that you can send to VR-Forces.

- "Chapter A. Troubleshooting" on page 313 explains how to resolve some common problems that can arise with VR-Engage.
- "Chapter B. Embarkation Slot Configuration in VR-Engage" on page 317 provides detailed information about configuring slots on vehicles to allow players to embark on them.

Documentation Set

VR-Engage documentation is provided as manuals in PDF format and online help. Electronic versions of the documentation are in *vrengage/doc*. On Windows, you can access the documentation from the VR-Engage folder on the Start menu. You can also find the most current documentation at *https://www.mak.com/support/product-user-guides*. The VR-Engage documentation set is as follows:

- *VR-Forces Users Guide* describes how to use the VR-Engage applications.
- *VR-Engage First Experience Guide*. The First Experience Guide provides a brief guided tour of the most important VR-Engage features. It is primarily for new users and persons evaluating VR-Engage.
- Online help. The online help, accessible from the Help menu, replicates most of the information in *VR-Forces Users Guide*.
- *VR-Vantage Users Guide*. The VR-Engage startup view gives you access to many settings that affect the visual quality of your role-playing experience. These and other aspects of the graphical user interface are shared with VR-Vantage and VR-Forces. *VR-Vantage Users Guide* is included to provide documentation for display settings, navigation of the GUI, and other information that may be of use to VR-Engage users, but which is not central to using VR-Engage.
- *Adding Content to MAK Applications Reference Manual* explains how to add visual models and terrain to VR-Engage.
- *VR-Engage Release Notes* lists system requirements, release-specific requirements, new features and updates, bug fixes, and known problems.

Try the Online Help

MAK's online help is the best way to use our documentation. It's easy to navigate, with improved natural language search tools that include filters to reduce long lists of search results. You can click the illustrations to see high-resolution enlargements that show more detail. And it's built into the product — just press **Shift**+**F1** to launch the help in your browser. You do not need to be connected to the internet to use it.

Use **F1** to access the list of game controller and keyboard controls for the roles in VR-Engage.

MAK Products

The MAK Technologies product line includes the following:

- VR-Link® Network Toolkit. VR-Link is an object-oriented library of C++ functions and definitions that implement the High Level Architecture (HLA) and the Distributed Interactive Simulation (DIS) protocol. VR-Link has built-in support for the RPR FOM and allows you to map to other FOMs. This library minimizes the time and effort required to build and maintain new HLA or DIS-compliant applications, and to integrate such compliance into existing applications.

  VR-Link includes a set of sample debugging applications and their source code. The source code serves as an example of how to use the VR-Link Toolkit to write applications. The executables provide valuable debugging services such as generating a predictable stream of HLA or DIS messages, and displaying the contents of messages transmitted on the network.

- MAK RTI. An RTI (Run-Time Infrastructure) is required to run applications using the High Level Architecture (HLA). The MAK RTI is optimized for high performance. It has an API, RTIspy®, that allows you to extend the RTI using plug-in modules. It also has a graphical user interface (the RTI Assistant) that helps users with configuration tasks and managing federates and federations.

- VR-Forces®. VR-Forces is a computer generated forces application and toolkit. It provides an application with a GUI, that gives you 2D and 3D views of a simulated environment.

  You can create and view entities, aggregate them into hierarchical units, assign tasks, set state parameters, and create plans that have tasks, set statements, and conditional statements. You can simulate using entity-level modeling, which focuses on the actions of individual people and vehicles, and aggregate-level modeling, which focuses on the interaction of large hierarchical units.

  VR-Forces also functions as a plan view display for viewing remote simulation objects taking part in an exercise. Using the toolkit, you can extend the VR-Forces application or create your own application for use with another user interface.

- VR-Vantage™. VR-Vantage is a line of products designed to meet your simulation visualization needs. It includes VR-Vantage and the VR-Vantage Toolkit.

  - VR-Vantage displays a realistic, 3D view of your virtual world, a 2D plan view, and an exaggerated reality (XR) view. Together these views provide both situational awareness and the big picture of the simulated world. You can move your viewpoint to any location in the 3D world and can attach it to simulation objects so that it moves as they do.

  - SensorFX. SensorFX is an enhanced version of VR-Vantage that uses physics based sensors to view terrain and simulation object models that have been materially classified. It is built in partnership with JRM Technologies.

  - The VR-Vantage Toolkit is a 3D visual application development toolkit. Use it to customize or extend MAK's VR-Vantage applications, or to

integrate VR-Vantage capabilities into your custom applications. VR-Vantage is built on top of OpenSceneGraph (OSG). The toolkit includes the OSG version used to build VR-Vantage.

- MAK Data Logger. The Data Logger, also called the Logger, can record HLA and DIS exercises and play them back for after-action review. You can play a recorded file at speeds above or below normal and can quickly jump to areas of interest. The Logger has a GUI and a text interface. The Logger API allows you to extend the Logger using plug-in modules or embed the Logger into your own application. The Logger editing features let you merge, trim, and offset Logger recordings.
- VR-Exchange™. VR-Exchange allows simulations that use incompatible communications protocols to interoperate. For example, within the HLA world, using VR-Exchange, federations using the HLA RPR FOM 1.0 can interoperate with simulations using RPR FOM 2.0, or federations using different RTIs can interoperate. VR-Exchange supports HLA, TENA, and DIS translation.
- VR-TheWorld™ Server. VR-TheWorld Server is a simple, yet powerful, web-based streaming terrain server, developed in conjunction with Pelican Mapping. Delivered with a global base map, you can also easily populate it with your own custom source data through a web-based interface. The server can be deployed on private, classified networks to provide streaming terrain data to a variety of simulation and visualization applications behind your firewall.
- DI-Guy™. DI-Guy is an SDK that allows you to embed the DI-Guy library in your real-time application and populate your world with lifelike human characters. DI-Guy provides high-fidelity human characters that move realistically, respond to simple high-level commands, and travel throughout the environment as directed by the hosting visual application. It comes with thousands of ready-to-use characters, appearances, and motions. DI-Guy character simulation is an integral part of all MAK's visual applications, including VR-Engage, VR-Forces, and VR-Vantage.
- RadarFX. RadarFX is a client-server application that simulates synthetic-aperture radar (SAR). The server application, which is based on VR-Vantage and SensorFX, loads a terrain database and, optionally, connects to simulations. A client application requests SAR images from the server. RadarFX includes a sample client application.
- VR-Engage. VR-Engage is a multi-role virtual simulator that lets users play the role of a first person human character, a ground vehicle driver, gunner or commander, a sensor operator, or the pilot of a fixed wing aircraft or helicopter. It incorporates the VR-Force simulation engine and the VR-Vantage graphics rendering capabilities.
- WebLVC Server. WebLVC Server implements the server side of the WebLVC protocol so that web-based simulation federates can participate in a distributed simulation. It is part of the WebLVC Suite, which includes the server and several sample applications that work with VR-Forces and VR-Vantage.
- MAK Legion. MAK Legion is a collection of applications and an SDK designed to support simulations with millions of simulation objects.

How to Contact Us

For MAK ONE technical support, information about upgrades, and information about other MAK products, you can contact us in the following ways:

Telephone

Call or fax us at:

Voice: 617-876-8085 (extension 3 for support)
Fax: 617-876-9208

E-mail

Sales and upgrade information: info@mak.com
Technical support: support@mak.com

Internet

MAK web site home page: www.mak.com
License key requests: www.mak.com/support/get-licenses
Technical support: jira.mak.com/servicedesk/customer/portal/1
Product version and platform information: www.mak.com/support/product-versions
For the free, unlicensed MAK RTI: www.mak.com/support/bonus-material

Post

Send postal correspondence to:

MAK Technologies
10 Fawcett Street, Suite 204
Cambridge, MA 02138 USA

When requesting support, tell us the product you are using, the version, and the platform on which you are running. If you are using VR-Vantage, VR-Forces, VR-Engage, or DI-Guy, please also include your data compatibility version and date.

Document Conventions

This manual uses the following typographic conventions:

Sans Serif Bold	Indicates a key on the keyboard or menu options. Also indicates parameters.
`Monospaced`	Indicates values you enter such as commands or arguments.
`Monospaced Italic`	Indicates command variables that you replace with appropriate values.
{}	Indicates required arguments.
[]	Indicates optional arguments.
|	Separates options in a command where only one option may be chosen at a time.
`(	)`	In command syntax, indicates equivalent alternatives for a command-line option, for example, `(-h | --help)`.

/	Indicates a directory. Since MAK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames.
Italic	Indicates a file name, pathname, or a class name.
➤	Indicates a procedure.
Menu > Option	Indicates a menu choice. For example, an instruction to select the Save option from the File menu appears as: Choose **File > Save.**
	Indicates supplemental or clarifying information.
	Indicates additional information that you must observe to ensure the success of a procedure or other task.

Directory names are preceded with dot and slash characters that show their position with respect to the MAK ONE home directory. For example, the directory *./vrengage2.0.1/doc* appears in the text as *./doc*.

For anything in the data directory, the path is shown starting with *./SharedData/16/latest* to be clear that it is not in the MAK ONE home directory. That is the default installation path. If you installed the data in a different directory, substitute your path.

Mouse Button Naming Conventions

An instruction to click the mouse button refers to clicking the primary mouse button, usually the left button for right-handed mice and the right button for left-handed mice. The context-sensitive menu, also called a popup menu or right-click menu, refers to the menu displayed when you click the secondary mouse button, usually the right button on right-handed mice and the left button on left-handed mice.

Third Party Licenses

MAK software products may use code from third parties. This section contains summary license information and where required, specific license documentation required by these third parties.

The following libraries are covered by the GNU Lesser General Public License (LGPL) license:

- CIGI
- DevIL
- GEOS
- GStreamer
- LibIconV
- LibWebSockets
- OPCODE
- OpenAL
- OSG
- OsgEarth
- Pthread
- Qt
- Qwt

The following libraries are covered by the ZLib license:

- Bullet Physics
- LibPNG
- LibSDL
- Minizip
- Zlib

The following libraries are covered by the MIT license:

- Expat
- FreeGLUT
- GDAL
- GeoTIFF
- GLM
- HarfBuzz
- LibCURL
- LibSquish
- LibXML
- LUA
- LuaBind
- Proj
- SMHasher
- TCLAP
- Qt Visual Graph Editor
- ZeroBrane Studio

The following libraries are covered by the BSD license:

- FreeType
- GLEW
- JPEG
- LibTiff
- Protobuf
- PCRE
- Ogg Vorbis

The following libraries are covered by the commercial licenses:

- FlexLM
- Gameware
- GLStudio
- JRM Technologies SenSimRT and SigSimRT
- OpenFlight
- SpeedTree
- Sundog Triton
- Sundog SilverLining
- CDB API

The following libraries are covered by custom licenses:

- NVidia CG Toolkit
- FreeImage (FreeImage Public License)
- Boost (Boost Software License)
- Poco (Boost Software License)

COLLADA DOM is covered by the SCEA Shared Source license.

Thread Building Blocks is covered by the GPL license:

SQLLite is in the public domain.

SWIG is covered by the GPL license.

Boost License

VR-Link, and all MAK ONE products that use VR-Link, uses some code that is distributed under the Boost License. All header files that contain Boost code are properly attributed. The Boost web site is: www.boost.org.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE

DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

libXML and libICONV

VR-Link, and all MAK ONE products that use VR-Link, links in libXML and libICONV. On some platforms the compiled libraries and header files are distributed with MAK products. MAK has made no modifications to these libraries. For more information about these libraries, see the following web sites:

- The LGPL license is available at: http://www.gnu.org/licenses/lgpl.html.
- Information about IconV is at: http://www.gnu.org/software/libiconv.
- Information about LibXML is at: http://xmlsoft.org.



1. Introduction to VR-Engage

This chapter is a brief introduction to VR-Engage.

1.1. VR-Engage Features 26
1.1.1 VR-Engage Versions 26
1.1.2 Simulation (Role-Playing) Capabilities 26
1.1.3 User Interface 31
1.1.4 Image Generator 31
1.1.5 Terrain 32
1.2. Integration with VR-Forces 32
1.3. Integration with Other MAK Products 33
1.3.1 After Action Review – Using MAK Data Logger 33
1.3.2 Network Interoperability with Other Federates 33
1.4. Extensibility 33
1.5. Licensing 33
1.5.1 Vortex License (for Ground Vehicle Dynamics) 34
1.5.2 RTDynamics License (For Aircraft) 34
1.5.3 License Requirements for Hosting a VR-Forces Session 34

1.1. VR-Engage Features

VR-Engage was developed for use in training simulations or laboratory experimentation. It lets users play the role of a first person human character, a vehicle driver, gunner or commander, a sensor operator, or the pilot of an airplane or helicopter.

VR-Engage is built on mature proven technologies. It gets its simulation engine from VR-Forces, its game-quality 3D graphics from VR-Vantage, and its network interoperability from VR-Link.

VR-Engage includes:

- A high-fidelity vehicle physics engine needed for accurate vehicle motion.
- Ground, rotary and fixed-wing vehicles, and the full library of friendly, hostile, and neutral DI-Guy characters.
- Radio and voice communications over DIS and HLA.
- Sensors, weapons, countermeasures, and behavior models for air-to-air, air-to-ground, on-the-ground, and person-to-person engagements.
- Vehicle- and person-specific interactions with the environment (open and close doors, move, destroy, and so on.)
- Terrain agility. As with VR-Vantage and VR-Forces, you can use the terrain you have or take advantage of innovative streaming and procedural terrain techniques.

1.1.1 VR-Engage Versions

VR-Engage has the following run-time versions:

- VR-Engage. Allows you to control air, ground, and human entities.
- VR-Engage with SensorFX. Includes the SensorFX plug-in, which gives you access to more sophisticated infrared and night vision settings.

Both configurations are installed as part of the standard VR-Engage installation process. Your access to them depends on your VR-Engage license. For simplicity, the procedures in this manual assume you are using VR-Engage.

1.1.2 Simulation (Role-Playing) Capabilities

VR-Engage lets you choose which entity you want to simulate. Said another way, you choose the role you want to play. You can play:

- Dismounted soldiers and other human characters.
- A member of the crew of a ground vehicle.
- A helicopter or fixed-wing aircraft pilot.
- The operator of a gimbaled sensor on a UAV or a mounted camera.

Dismounted Soldiers and Other Human Characters

If you take the role of a dismounted soldier or other human character, you have access to the full DI-Guy library of friendly, hostile, and neutral characters. Your character has the following capabilities:

- Individual movement (look, run, walk, crawl, crouch, prone).
- Interact with environment (open and close doors).
- Interact with other characters (shoot at and be shot at).
- Sensors (binoculars and night vision goggles (NVG)).
- Weapons (rifles, grenades).
- Use carried gear, such as a flashlight, range finder, or compass.
- Lase targets for other players or simulated entities.
- CGF Assist (passenger, fast-roping). When you are running a VR-Forces scenario, you can give up control of your character and let the simulation engine manage complex behaviors.
- Embark and disembark. Enter and leave vehicles and take on roles of vehicle crew members.

Figure 1. Dismounted infantry



Ground Vehicle Crew

VR-Engage allows you to take on the roles of the driver, gunner, or commander of a ground vehicle that has the following capabilities:

- Physics-based ground vehicle dynamics model.
- Interact with environment (destroy).
- Ground-to-ground engagements:
  - Sensors (EO, IR, NVG).
  - Weapons (main gun, machine gun).
  - Signals/Countermeasures (smoke grenades).
- Vehicle controls (headlights, tail lights).
- CGF Assist (drive, gun). In a VR-Forces scenario, as you switch roles, the simulation engine can take over the tasks and plans that manage the other roles for the vehicle. For example, if you are the gunner, VR-Forces can manage driving the vehicle.

Figure 2. Bradley gunner



You can easily switch roles and also have the ability to disembark from the vehicle to assume the role of a dismounted soldier.

Pilot

As a pilot, you can fly an aircraft with the following capabilities:

- Physics-based fixed & rotary wing dynamics models.
- Multi-function cockpit display. Flight instruments, map/chart, imagery.

- Air-to-Air Engagements:
  - Sensors (radar scope, radar warning receiver).
  - Weapons (missiles).
  - Countermeasures (chaff, flares).
- Air-to-Ground Engagements:
  - Sensors (IR targeting, SAR (requires RadarFX Server)).
  - Weapons (missiles, guns, bombs).
  - Countermeasures (chaff, flares).

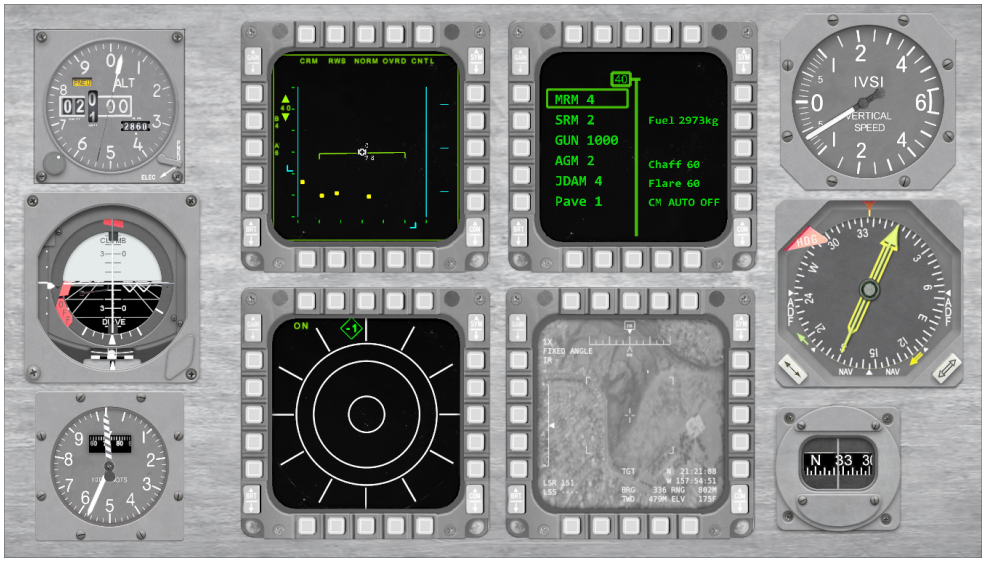
Pilots use their out-the-window views and head-up display (HUD) (Figure 3) to understand their environment and control their aircraft.

- Single screen. Out-the-window view, on-screen controls, and HUD.
- Dual screen. Out-the-window view, on-screen controls, and HUD.
- On-screen multi-function display (IR, SAR (requires SensorFX)) (Figure 4).
- Slaved IG (requires VR-Vantage Remote Display Engine).
- Head-mounted display (HMD). Oculus Rift, HTC Vive, or Varjo XR-1 and VR-2.

Figure 3. Head-up display with instruments



Figure 4. Instrument panel



Sensor Operator

In the sensor operator role, you can perform common surveillance and reconnaissance tasks, such as tracking fixed and moving targets, using a simulated E/O camera or IR sensor. You can control gimbaled sensors using joysticks or game controllers or configure VR-Engage to work with sensor-specific hand controller devices. VR-Engage lets you attach cameras to entities and structures and set their initial heading and pitch. VR-Engage can also automatically attach cameras to remotely simulated (that is, non-VR-Forces) entities.

Figure 5. Sensor view



Drone Operator

As a drone operator, you can deploy a small drone equipped with a camera and then fly it around. This role is designed to train people how to safely fly a drone without risking an actual device. You can use these in many different types of scenarios, such as simple infrastructure inspection or surveillance and reconnaissance.

Figure 6. Drone operator



1.1.3 User Interface

You can control your character using the keyboard, mouse, and game controllers. The graphical user interface provides command menus and cockpit and crew cabin displays. Other interfaces include:

- Radio (push-to-talk, radio channels, MAK Data Logger support).
- Control with role-specific devices (driver's steering wheel and pedals, pilot's HOTAS).
- Display system support (monitors, projectors, mixed and virtual reality headsets).
- Audio (engine sounds, environmental noise, event effects, warning tones).

1.1.4 Image Generator

VR-Engage includes a built-in image generator, based on VR-Vantage, that can support multiple display channels, to the limits of your installed graphics card. It can also be extended to fill larger multi-channel display systems by adding VR-Vantage Remote Display Engines. This is an example of the flexibility of MAK's product architecture that allows you to configure a system of any size.

Regardless of which role your VR-Engage simulator is playing, VR-Engage delivers realistic scene rendering (terrain, entities, interaction effects, weather, clouds, time-of-day, lighting, shadows, high dynamic range (HDR) lighting.)

The image generator supports the following configurations:

- Single screen.
- Dual screen.
- Multi-channel distributed rendering (requires VR-Vantage Remote Display Engine). Spread the view across multiple monitors run by multiple computers.
- IR and NVG Sensors using the included CameraFX or optional SensorFX.
- On screen Multi-function Display (IR, SAR (requires SensorFX)).
- Head-mounted displays.



When you use remote display engines with VR-Engage, you must connect to any remote display engine before hosting or joining a session. Attempting to connect to a remote display engine after a session has started, VR-Engage leaves the session and then ends up in a state where it constantly makes unsuccessful attempts to rejoin the session.

1.1.5 Terrain

VR-Engage, like VR-Vantage and VR-Forces, allows you to use the terrain you have or take advantage of innovative streaming and procedural terrain techniques to simulate on large, complex terrains.

- Terrain agile (all types supported by VR-Vantage and VR-Forces).
- Dynamic terrain. Destroyable buildings, bridges, and props.
- Dynamic ocean. Waves, spray, wakes, wind.

VR-Engage includes a set of useful geo-specific terrains and options for building more. See the *Adding Content to MAK Applications Reference Manual* for information about terrain support. (The *Adding Content to MAK Applications Reference Manual* is included with VR-Engage.)

1.2. Integration with VR-Forces

When VR-Engage is used with a compatible version of VR-Forces and other MAK products, you can reap the additional benefits of a common system architecture:

- Common representation of the environment across player and CGF stations, including synchronized weather, time-of-day, and dynamic terrain.
- Build terrains, models, and configurations once, and deploy them across VR-Engage player stations, VR-Forces simulation engines and front-ends, and any other applications that use VR-Vantage.
- Role-play multiple entities at a time by switching between manual and CGF control on-the-fly. Take control of VR-Forces-driven entities.
- Configure players with the VR-Forces Simulation Object Editor.

- Create scenarios in VR-Forces and load them in VR-Engage for standalone use.
- Automatic WebLVC integration (tablet-based IOS, and so on.)

1.3. Integration with Other MAK Products

Previous sections have discussed VR-Engage's integration with VR-Forces and VR-Vantage. Additionally, like all MAK products, VR-Engage uses VR-Link for network interoperability and supports DIS and HLA. It integrates seamlessly with our other Link products, such as MAK Data Logger for recording and AAR, the MAK RTI for HLA support, and VR-Exchange for data interchange in heterogeneous simulation environments.

1.3.1 After Action Review – Using MAK Data Logger

Whether in standalone mode or when playing with other simulation applications, VR-Engage publishes to the DIS or HLA network. Therefore, you can use the MAK Data Logger to record your simulation and play the recording back for after-action-review (AAR).

1.3.2 Network Interoperability with Other Federates

VR-Engage is natively compliant with DIS and HLA, allowing it to participate in any topology of networked simulators and simulation support systems, which includes multi-player classroom environments. It can interoperate with existing simulation applications and 3rd party simulators, SAFs, and CGFs.

1.4. Extensibility

VR-Engage is ready to use out-of-the-box. It can be deployed as a trainee simulator, as a role player station, an instructor aid, a desktop simulation game, or even as a VR headset experience.

Because MAK focuses on the needs of system integrators, we designed flexibility into VR-Engage. It can be customized and extended to meet program-specific requirements, and can be integrated into a diverse range of system configurations. MAK has a pricing and support model that has proven to be effective with system integrators both before and after they choose MAK.

1.5. Licensing

The VR-Engage application requires a VR-Engage run-time license in order to run. Internally, VR-Engage relies on VR-Vantage capabilities for its 3D visualization, VR-Forces capabilities for its ownship simulation, VR-Link for its networking, and so on. These capabilities are part of VR-Engage and do not require separate run-time licenses of these MAK products.

> You can run a limited version of VR-Engage—air, ground, or sensor—using a command-line option, as described in "3.13. Command-Line Options" on page 72. If you only have a license for one of these limited versions and you run the full version of vrEngage.exe without one of the command-line options, you can see other features but not use them.

1.5.1 Vortex License (for Ground Vehicle Dynamics)

VR-Engage uses Vortex from CM Labs for its ground vehicle dynamics. This built-in runtime capability is included with VR-Engage. You do not need a separate license to run VR-Engage.

If you want to use the Vortex Editor to edit VR-Engage's built-in vehicle dynamics models or author new ones must purchase a copy of Vortex Studio directly from CM Labs. The right to execute the resulting custom content in VR-Engage is covered by the VR-Engage license, and does not require a separate Vortex runtime license.

> For more information, see "13.2. Ground Vehicles and Vortex" on page 276.

1.5.2 RTDynamics License (For Aircraft)

VR-Engage uses the RotorLib and FixedWingLib products from RTDynamics for its aircraft vehicle dynamics. This built-in runtime capability is included with the VR-Engage license. You do not need a separate RotorLib or FixedWingLib license key to run VR-Engage.

If you want to use RTDynamics's RIME editor to edit VR-Engage's built-in flight dynamics models or author new ones must purchase a copy of RIME or the RTDynamics SDKs. You can purchase these through MAK; contact your account representative or sales@mak.com. The right to execute the resulting custom content in VR-Engage is covered by the VR-Engage license, and does not require a separate Vortex runtime license.

1.5.3 License Requirements for Hosting a VR-Forces Session

With only a VR-Engage license, you can run VR-Engage, choose a terrain database to simulate on, create and place an entity, and control that entity. VR-Engage can only simulate and control a single entity at a time. When you change roles and create a different entity to simulate, the entity you were previously simulating is deleted and replaced by the new one.

If you are running a copy of VR-Forces on your network (which requires a VR-Forces license, of course), then VR-Engage can interact with VR-Forces:

- VR-Engage allows you to join a VR-Forces session and take manual control of one of the entities in the VR-Forces scenario. If you are controlling a VR-Forces entity, and then change roles and ask to control a different entity, the original entity will still exist, as it is owned by VR-Forces.

- VR-Engage can initiate and receive changes to the terrain, for example, opening doors or destroying buildings. (VR-Forces acts as the dynamic terrain server.)
- VR-Engage can reflect changes to time-of-day and weather that are published by VR-Forces.
- VR-Engage allows you to assign certain tasks to entities that are simulated by VR-Forces (for example, tell a subordinate to follow you, or move to a location you designate).

In order to simplify the startup workflow for certain use cases, VR-Engage gives you the ability to launch a VR-Forces back-end directly from the VR-Engage user interface, and to instruct VR-Forces to load and host a scenario of your choosing. This allows you to avoid separately launching VR-Forces from the start menu or command line, and automatically ensures that VR-Forces uses the same interoperability parameters as VR-Engage. A VR-Forces back-end license is always required to run VR-Forces, whether you launch it from the VR-Engage user interface, or launch it separately from the Start Menu or command line. If you do not have a VR-Forces back-end license available, VR-Engage will not allow you to launch VR-Forces and host a VR-Forces scenario.

When VR-Engage starts up, it looks for a running VR-Forces and, if it finds one, it connects to it. If it does not find one, it offers you the option to host and load a VR-Forces scenario or start an unhosted session. If you host a VR-Forces scenario, you consume a VR-Forces license, which prevents you from starting VR-Forces independently if you have only one license. If you start an unhosted session, you can start or join a session only as an individual player—you cannot load and run a scenario.



2. Installing VR-Engage

This chapter explains how to install the VR-Engage applications and data. You must also install the MAK license manager. To use VR-Engage with HLA, you must install an RTI.

2.1. Installing VR-Engage 38
2.1.1 Installing VR-Engage on Windows 38
2.1.2 Uninstalling VR-Engage from Windows 39
2.1.3 Installing the makData Package 39
2.2. Upgrading VR-Engage Simulation Model Sets 40
2.3. The VR-Engage Directory Structure 40
2.4. Installing and Setting Up the MAK License Manager 41
2.4.1 Specifying the License Server 41
2.5. Installing an RTI 43
2.5.1 Installing the MAK RTI 44
2.6. Input Devices 45
2.6.1 HMD Support 45
2.7. Factors that Affect Performance 45
2.8. Localizing the Graphical User Interface 46

2.1. Installing VR-Engage

A full VR-Engage installation includes the application, data, the License Manager software and, for HLA operation, an RTI (or Runtime Infrastructure). If your license includes SensorFX, that is installed automatically with VR-Engage.

VR-Engage has an application installer and a data package (*.MAK-targz*) that has multiple files. You must run the application installer and then install the data package.

Before you install VR-Engage, read the release notes to see if there are any special instructions for installation.

2.1.1 Installing VR-Engage on Windows

When you install large applications on Windows, there may be a delay of up to several minutes from the time you try to run the setup program to the time that an installation dialog box is displayed. This is due to how Windows scans setup programs before it executes them. If you experience this problem, turning off User Access Control can reduce or eliminate this delay.

Windows versions of VR-Engage are provided as executable installer files on DVD, or as downloaded files. The installers are named to indicate the compiler used to build that version of VR-Engage.

➤ To install VR-Engage, run the application installer. Follow the instructions in the installation wizard.

Default Windows Installation Directory

On Windows, the default is that VR-Engage installs in the *C:/MAK* directory. You can use another directory by defining a MAK_INSTALL_DIR environment variable. If you set up this environment variable, the product installs in the directory you specify instead of *C:/MAK*.

Installing the VR-Forces Plug-in

When the VR-Engage installer completes the data installation, it informs you of the plug-in that is available for VR-Forces. The VR-Forces plug-in allows you to take control of VR-Forces entities from VR-Engage if they are configured to be controlled by VR-Engage. If you have VR-Forces installed and want to install the plug-in, run the *VR-ForcesVR-EngageAdd-ons.exe* installer. This is packaged separately from the VR-Engage installer.

The VR-Forces plug-in makes the following changes to VR-Forces:

- Installs the *./data/simulationModelSets/VR-Engage.sms* and custom models.
- Replaces some VR-Forces level data or configuration files. In those cases, it saves the original version is preserved by the installer by copying it to <*original file name*>.*vrengage-addons-backup*. If you uninstall the VR-Engage add-ons installer, any VR-Forces files that the installer modified are restored by copying those **.vrengage-addons-backup* files to their original names.
- Adds RTDynamics and VREngage simulation model sets.
- Adds the VR-Engage revision to the About dialog box.

You must install the *VR-ForcesVR-EngageAdd-ons.exe* plug-in on top of a compatible version of VR-Forces. The *VR-ForcesVR-EngageAdd-ons.exe* installer attempts to check that the version of VR-Forces is compatible with your version of the VR-Engage add-ons. You can also refer to the VR-Engage release notes to learn which version of VR-Forces you can use with your version of VR-Engage. If you install the plug-in on an incompatible version of VR-Forces, it is not guaranteed to run reliably or correctly (if at all).

2.1.2 Uninstalling VR-Engage from Windows

MAK Technologies product installations include an uninstaller application, which you can find in the installation directory. When you uninstall VR-Engage, the uninstaller does not remove the data package; that is in a separate directory (*./SharedData/16/latest*). Files such as scenarios or SMSs that reside in the product directory are deleted when you uninstall VR-Engage. Be sure to copy any custom files from the product directory before uninstalling.

2.1.3 Installing the makData Package

As of the VR-Engage 2.0 release, you no longer need to install the data separately for each MAK product. Using the MAK Software Manager, you install the data in a central location and then MAK products can share it.

The default location for the shared data files is *C:/MAK/SharedData/16/latest*. You can install the files somewhere else, but in that case you must edit the MAK_SHARED_DATA_ROOT and MAK_SHARED_DATA_STEM environment variables or use the command line option `--sharedDataRoot` to specify the location of the directory.

If you do not install the package and you open VR-Engage, an error message displays: "Data package not found in specified data directory: '*directory path*'. Please make sure the data package is installed." The *directory path* is the either the default location or the one that you specified.

➤ **To install the data:**

1. Install the MAK ONE Software Manager.
2. Run the MakOneSoftwareManager executable in the *.\bin64* folder.

   On Windows 10, you can also open the Start menu and then choose **MAK Technologies > MAK ONE Software Manager**.

3. Click Select and Install Data Packages in the welcome page.
4. Specify the directory where you saved the data files. The default is your Downloads folder, but you can click the folder icon to navigate to another folder.
5. Select the data files that you want to install.

> You must install the Core data for VR-Engage to function. To have access to other terrains and to see anything other than notional models, install the other data packages provided by MAK Technologies. If you do not install any other data, it is assumed that you are providing your own models and terrains.

6. Click Install. The MAK ONE Software Manager shows the progress of the data installation.
7. When the installation is complete, click Exit.

2.2. Upgrading VR-Engage Simulation Model Sets

When you upgrade to new, compatible versions of VR-Engage and VR-Forces, you must upgrade your simulation model set, as well. Refer to the documentation for the new version of VR-Forces, specifically the *VR-Forces Migration Guide* and "Upgrading Old Simulation Model Sets" in the *VR-Forces Users Guide*.

2.3. The VR-Engage Directory Structure

The installation process creates the VR-Engagedirectory structure shown in Figure 7.

Figure 7. VR-Engage directory structure

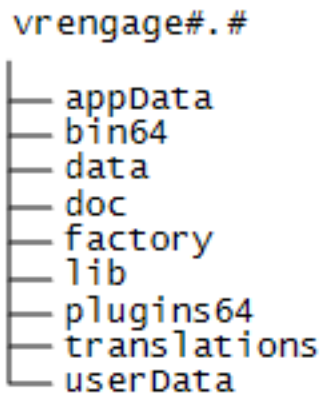


Table 1 describes the contents of the directories.

Table 1. Contents of VR-Engage directories

Directory	Directory contents
./appData	Subdirectories contain data required by the VR-Engage applications, such as model definitions and saved settings. Also contains data required by the SensorFX plug-in for VR-Engage.
./bin64	Executables and dynamic link libraries.
./data	Sample databases, images, and vehicle models.
./doc	VR-Engage documentation and class reference files.
./factory	Backups of the application data and settings provided by MAK. Also includes the default settings for SensorFX (for reverting to "factory" settings).
./lib	Libraries for Tracy client.
./plugins64	Plug-in shared libraries.
./translations	Translation files for localizing VR-Engage menus.
./userData	Terrain and scenario files.

2.4. Installing and Setting Up the MAK License Manager

Before you can use a MAK product, you must obtain a valid license file, install the MAK License Manager, and configure the license server and client machines. The License Manager uses a client-server architecture, so you do not need to install the License Manager on every computer on which you install MAK products. You only need to install it on the computer that you will use as the license server.



If you have already installed the License Manager for another MAK product, you do not have to install it again. You just need to make sure you have licenses for your newly installed products.

The License Manager installer is included on MAK installation media. It is separate from the product installers. You can download the installers from our web site at *https://www.mak.com/support/license-support*.

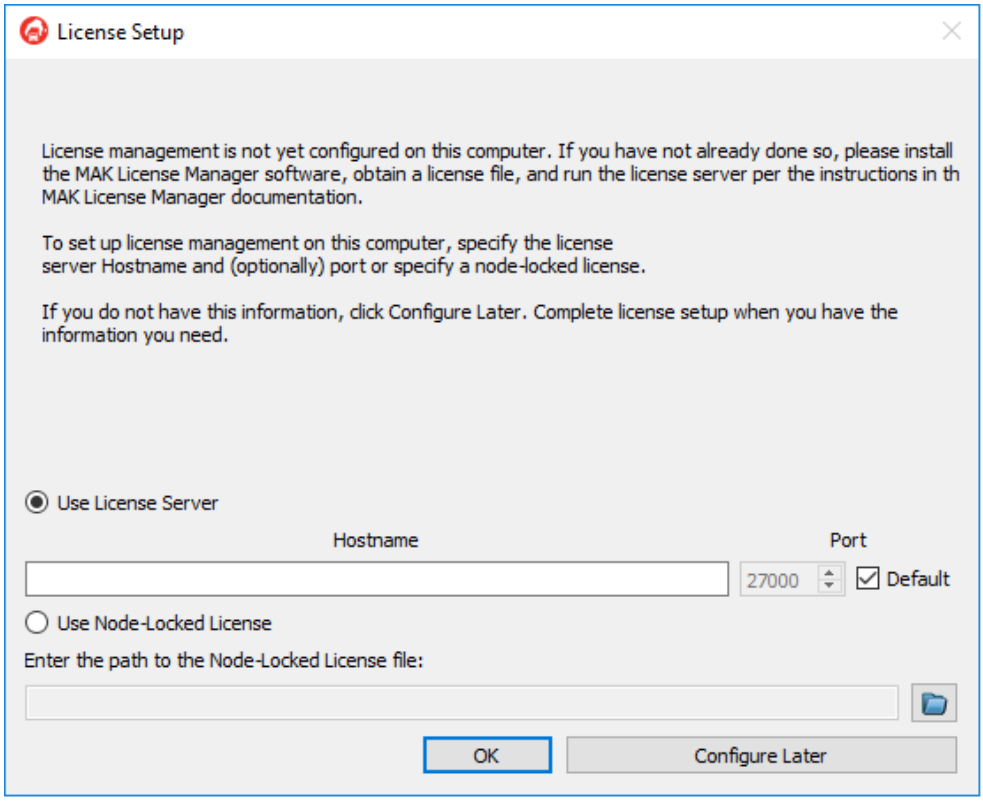
Complete installation and configuration instructions are included with the License Manager installer. Instructions are also available at the license support page.

Some customers use dongle licenses instead of running a license server. Instructions for using dongles are in the License Manager documentation.

2.4.1 Specifying the License Server

The first time you run a MAK application on a particular computer, the License Setup dialog box opens (Figure 8). It prompts you to enter the hostname of the license server and optionally, a port number.

Figure 8. License Setup dialog box



If you do not know the hostname of the license server, click Configure Later. When you have the hostname, you can start the application again and complete the dialog box. You will not be able to run any MAK applications until you set up license management.

If you know the hostname, type it in the Hostname box and then click OK. The application will start.

Under limited circumstances, MAK issues node-locked licenses. If you have a node-locked license, select the Use Node Locked License option and enter the path to the license file.

- If you are running MAK products on the license server machine, it is also a client, so you must specify the license server on that machine, too.
- If you change the license server, the saved configuration will no longer be valid and the License Setup dialog box will open the next time you start a MAK application.
- You can clear the saved license configuration by deleting the cache file. On Windows, it is *C:\Users\user_name\AppData\Roaming\MAK\licenses<n>.txt*. (The *AppData* directory is hidden by default.) On Linux, it is *.mak/license<n>.txt*. (There may be more than one cache file, for example, *licenses1.txt* and *licenses2.txt*.) You can also clear the cache in the RTI Assistant.

The MAKLMGRD_LICENSE_FILE Environment Variable

As an alternative to using the License Setup procedure described in the previous section, you can configure the license server in an environment variable. The MAKLMGRD_LICENSE_FILE environment variable identifies the server machine. If you set this environment variable, it overrides the settings stored by the License Setup procedure.

The syntax for the environment variable is: *@Server_name*. For example, if the server machine is oak, set the environment variable to **@oak**.

➤ **To add the MAKLMGRD_LICENSE_FILE in Windows:**

1. On the Start menu, click the Settings button. The Settings window opens.
2. In the search box, type environment. A list of choices is displayed.
3. Select Edit the System Environment Variables. The System Properties dialog box opens.
4. Click Environment Variables. The Environment Variables dialog box opens.
5. In the System Variables group box, click New. The New System Variable dialog box opens.
6. In the Variable Name field, enter MAKLMGRD_LICENSE_FILE.
7. In the Variable Value field, enter *@server_name*, where *server_name* is the name of the license server.
8. Click OK to back out of each dialog box and set the variable.

2.5. Installing an RTI

An RTI is a software library (and perhaps supporting executables) that implements an HLA Interface Specification. In HLA, applications exchange FOM data through RTI calls, which means that all HLA applications need to use an RTI.



Because of differences in the low-level network mechanisms used by different RTI implementations (which include, but are not limited to packet layout), applications that want to interoperate in the same federation execution must use the same RTI implementation.

Because RTIs are usually provided as dynamic libraries that implement a fixed API, a federation can often switch from one RTI implementation to another between runs (without even recompiling the applications), but during each run, all participants must agree on which RTI to use, much as they must also agree on which FOM to use.

For the most recent information about the RTI versions supported by MAK products, please see the release notes for your MAK application.

2.5.1 Installing the MAK RTI

To install the MAK RTI, follow the instructions in Chapter 2 of *MAK RTI Users Guide*.

Configuring Your System to Use the MAK RTI

The RTI dynamic libraries must be located somewhere on your dynamic library search path. The path is indicated by an environment variable as follows:

- Linux — LD_LIBRARY_PATH
- Windows — PATH

The MAK RTI needs to know where to find the following configuration files:

- The federation configuration file (*.fed*, *.fdd*, or *.xml*) for your federation execution (required).
- The RID file (*rid.mtl*) (optional).

Put the configuration files in the directory from which you are running, or set the environment variable RTI_CONFIG to the directory that contains them.

Running Applications with the MAK RTI

➤ **To run a MAK application with the MAK RTI:**

1. Be sure the license server is running.
2. Start the application. The RTI Assistant will prompt you to choose an RTI configuration.
3. Choose a configuration. If necessary, start the rtiexec.
4. Click Connect. The application should run.

In many cases, you do not need to run the rtiexec to use the MAK RTI, however you can run the rtiexec if you want to. (It is required to use certain features of the MAK RTI.) For more information, please see your RTI documentation.

2.6. Input Devices

VR-Engage can support any device that Windows recognizes as a game controller. VR-Engage has tested the following game controllers and has Direct Input configurations set up for them:

- Xbox controller
- Logitech F310
- Logitech F710
- Logitech G27 Steering Wheel
- Logitech G29 Driving Force game steering wheel
- Thrustmaster Hotas Warthog
- Thrustmaster T-Flight Hotas X

The mappings for these devices and the keyboard are described in "Chapter 11. Mouse, Keypad, and Controller Mappings" on page 189. Please contact support (https://jira.mak.com/servicedesk/customer/portal/1) for assistance configuring other devices.

2.6.1 HMD Support

VR-Engage 2.0.1 supports a number of virtual and mixed reality devices. See "3.10. Introduction to Head-Mounted Displays" on page 64 and also the chapter *Using Head Mounted Displays for Virtual and Mixed Reality* in the *VR-Vantage Users Guide* to learn more.

2.7. Factors that Affect Performance

The performance of a visualization product such as VR-Engage depends on:

- Processor speed.
- Video card. (Always use the latest drivers for your video card.)
- Available memory.
- Size of the terrain database.
- The number and complexity of models loaded and how well they have been optimized for performance. For information about optimizing models, see *"Best Practices for Creating Models"* in the *Adding Content to MAK Applications Reference Manual*.
- Number of objects in the simulation and how active they are.
- Use of model instancing.



Nahimic audio software can have a negative impact on performance. See "A.1. Performance and Display Issues" on page 314 to learn how to add the application to the *BlackApps.dat* file.

2.8. Localizing the Graphical User Interface

You can localize your MAK Technologies product's graphical user interface so that it uses languages other than English. For details, see "Localizing the Graphical User Interface" in the *VR-Vantage Users Guide*.



3. Starting VR-Engage

This chapter describes how to start up VR-Engage, connect to a session, and choose a role.

3.1. Starting VR-Engage 49
3.1.1 Hosting a VR-Forces Session 49
3.1.2 Starting an Unhosted Session 50
3.1.3 Admin, User, and Lockdown Modes 51
3.2. Choosing a Role 52
3.2.1 Choosing an Existing Role 53
3.2.2 Creating a New Entity 54
3.2.3 Using an HMD with an Entity 55
3.2.4 Assigning Roles from VR-Forces 56
3.3. Connecting to VR-Forces 58
3.4. Changing Your Role 59
3.4.1 Changing Your Role in the Choose Role Panel 59
3.4.2 Changing Your Role by Embarking on a Vehicle 59
3.5. Entity Engagement and Scenario State 60
3.6. Overview of the Application Menu 61
3.7. Starting a New Session 62
3.8. Configuring the Simulation Connection 62
3.9. Playing in Windowed Mode 63
3.10. Introduction to Head-Mounted Displays 64
3.10.1 Prerequisites for Using an HMD 64
3.10.2 Resetting the Tracking Sensor 64
3.10.3 Calibrating the Eye Tracking for Varjo 65
3.10.4 Enabling and Disabling Virtual Reality Rendering 65
3.10.5 Configuration Options for Virtual and Mixed Reality 66
3.10.6 Setting Up Mixed Reality 67
3.10.7 Optimizing Performance with HMDs 70

3.11. Creating a Startup Configuration 70
3.11.1 Minion and Boss Configuration 71
3.12. Starting Multiple Simulation Engines 72
3.13. Command-Line Options 72
3.14. Command-Line Options for Player Stations 85
3.15. Managing Plug-ins 85
3.15.1 Loading Plug-ins 86
3.15.2 Adding a Plug-in 86
3.15.3 Specifying the DLLs for a Plug-in 87
3.15.4 Adding a Plug-in Configuration 88
3.15.5 Deleting a Plug-in Configuration 89
3.15.6 Deleting a Plug-in 89

3.1. Starting VR-Engage

When you start VR-Engage, it looks for an instance of VR-Engage that is already running or an instance of VR-Forces that is running, checking for a session to join. A session manages the relationship between the simulation engine, or back-end, and the GUI, or front-end. For more information, see the chapter *VR-Forces Application Concepts* in the *VR-Forces Users Guide*.

If VR-Engage finds a session, it immediately connects to it, loads the terrain it is using, and then asks you what character you want to control. If it does not find a session, it stops and asks you what to do. You can host a VR-Forces session or start an unhosted session.

You can set some of the startup behaviors with a startup configuration script. For details, see "3.11. Creating a Startup Configuration" on page 70.



If you join a session hosted by either VR-Forces or another instance of VR-Engage, your instance of VR-Engage does not create a sim engine by default.

If you want to run a simulation engine for your instance of VR-Engage, you must edit the *./appData/scripts/playerStation.lua* file and change the `alwaysRunOwnshipSim` parameter to `true`.

3.1.1 Hosting a VR-Forces Session

VR-Engage includes scenarios that were created using VR-Forces. When you host a VR-Forces session, you can load one of these scenarios or start an empty scenario on any installed terrain. If you run an existing scenario, you can control any of the entities in them. You can also add and control one other entity. If you start an empty scenario, you can only add and control one entity.



You cannot save scenarios in VR-Engage. The principal benefit of hosting a VR-Forces session is that a VR-Forces session supports global weather and dynamic terrain. If you want to create and save scenarios for use with VR-Engage, you must run VR-Forces with the VR-Engage add-ons.

If other instances of VR-Engage are running, they can connect to the session. Each instance can only control one entity.



When you host a VR-Forces session, you use a VR-Forces license. If you only have one VR-Forces license and you host a VR-Forces session from VR-Engage, you will not be able to start the VR-Forces application.

➤ **To start VR-Engage and host a VR-Forces session:**

1. On the Start menu, choose **MAK Technologies > VR-Engage**. The VR-Engage window opens.

2. In the Choose Connection screen, select the connection that you want to use and then click Connect. The Searching for a Session screen displays.
3. If VR-Engage finds a VR-Forces session, it connects to it.

   If it does not find a session, the Searching for Session screen displays (Figure 9).

Figure 9. Searching for Session



> By default, the VR-Engage window fills the screen, which we call full-screen mode. Admin mode shows title bar and menu bar typical of most windowed applications. We call this windowed mode. Full-screen mode does not have menus or other window controls. Although full-screen mode is the default for controlling characters, you can also control them in windowed mode. For details, see "3.9. Playing in Windowed Mode" on page 63.

4. Click Host VR-Forces Session. The Choose a Scenario for this Session dialog box opens, where you can load a scenario.
5. Select the scenario you want to load.
6. Click Load Scenario. A loading screen displays while opening the scenario and starting a sim engine. Once the terrain and scenario load, VR-Engage displays the scenario in 3D View mode and opens the Choose Role Panel (Figure 10 on page 52).
7. Choose a role, as described in "3.2. Choosing a Role" on page 52.

3.1.2 Starting an Unhosted Session

In an unhosted session, you cannot load scenarios. You can load an installed terrain and create a single entity on it to control. If there are other VR-Engage unhosted sessions running on the network, you can interact with them.

Unhosted sessions do not support dynamic terrain. Destructible buildings and buildings with switchable state windows or doors do not function in an unhosted session. As a result, players cannot damage buildings and cannot open and close doors or windows on buildings.

➤ **To start VR-Engage and host a VR-Forces session:**

1. On the Start menu, choose **MAK Technologies > VR-Engage**. The VR-Engage window opens.
2. In the Choose Connection screen, select the connection that you want to use and then click Connect. The Searching for Session screen displays (Figure 9 on the previous page).
3. Click Start Unhosted Session. The Choose Terrain list displays.
4. Select a terrain.
5. Click OK. VR-Engage displays the terrain in 3D View mode and the Choose Role Panel opens (Figure 10 on the next page).
6. Create an entity to control, as described in "3.2.2 Creating a New Entity" on page 54.

3.1.3 Admin, User, and Lockdown Modes

There are three modes for VR-Engage: Admin (administrator), User, and Lockdown:

- User mode, which is the default, is meant for role players that are allowed to set the DIS/HLA networking parameters, choose the sessions and entities that they engage with, and select their roles. This mode allows users to host a VR-Forces session (assuming you have a VR-Forces license), join an existing VR-Forces session, or add a new VR-Engage-owned entity to an exercise. Users can play and pause the scenario that they are hosting, or are participating in an unhosted scenario.

  User mode does not allow the player to access the Qt-based menus and toolbars that VR-Engage inherits from VR-Vantage to set advanced visualization and application settings. It also omits administrative functionality, such as Reset and Reset in Place. You can switch to Admin mode to access those functions.

- Admin mode is for developers and administrative staff, allowing them to configure, debug, customize, and tune the performance of VR-Engage. It includes all of the capabilities of User mode, and adds the ability to edit Display Engine configurations as well as access many of the VR-Vantage rendering engine's menus and toolbars. From those UI elements, you can control and monitor visuals such as shadows, high-dynamic range rendering (HDR), lighting, ocean settings, time and weather, and much more. Admin mode also allows you to reset and restore an entity you are controlling if it is destroyed or otherwise inoperable.

- Lockdown mode is intended for deployed training systems where the scenarios will be loaded and roles assigned by an instructor/operator using VR-Forces. It is more restrictive that User mode because it disables the Choose Session, Choose Role, and Create Entity functions. The player can only play the roles assigned to them through the instructor/operator station, scripts, or command-line options.

There are three methods for specifying the mode:

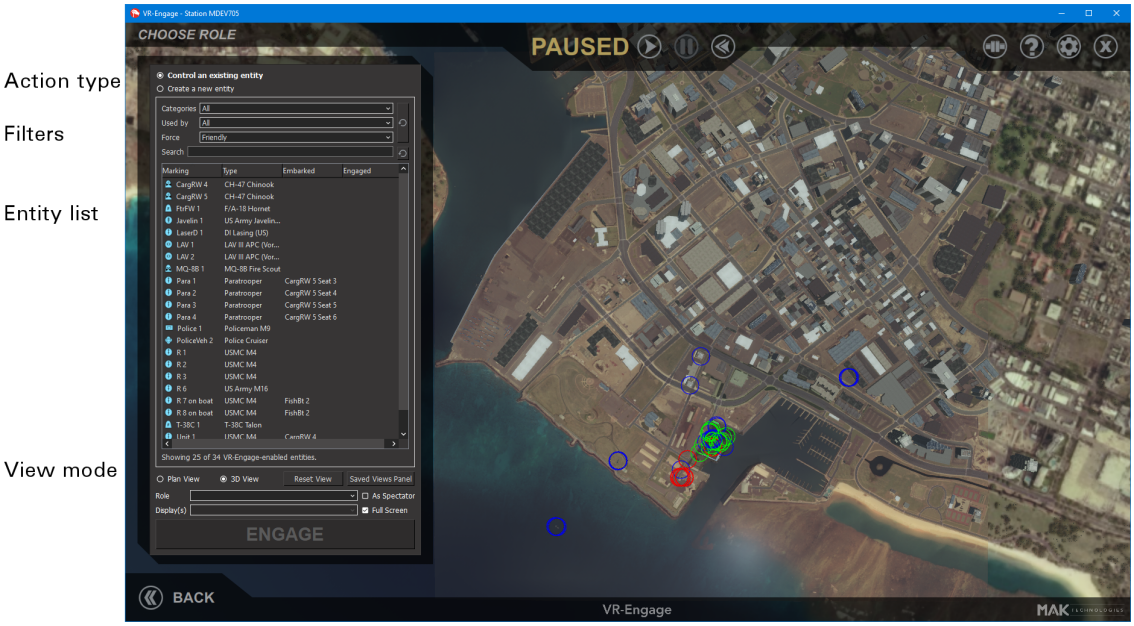
- Use the `--appMode` command-line option to specify which mode you want to use: `user`, `admin`, or `lockdown`.
- At runtime, toggle between the Admin and User modes by pressing **Shift**+**F12**. This does not work in Lockdown mode.
- Specify the mode to use in the *./appData/scripts/playerStation.lua* file's **applicationMode** parameter.

3.2. Choosing a Role

When you connect to or start a session, you can then choose a role to play. The VR-Engage window lists the available roles in the Choose Role Panel (Figure 10). If another player station is controlling a role, that is indicated in the Engaged column and you cannot take control of that role.

If you load a scenario or connect to a simulation, you can choose one of the entities in the simulation or create a new entity (except in lockdown mode). If you start an unhosted session, you must create a new entity and choose that entity to control.

Figure 10. Choose Role Panel



VR-Engage users typically control a character in the simulation. However, you can also participate as a spectator. A spectator attaches to an entity just like a player would, but cannot control the entity. The entity would be controlled by another player or by another application, such as VR-Forces.

When you host a session and engage in a role, the scenario begins to run. If you join a session hosted by VR-Forces or another instance of VR-Engage, the user of that application controls when the scenario is running or paused. If you create an unhosted session, the scenario does not begin to run until you click Play.



Multiple players should not try to control the same vehicle or person in the same role. If multiple players take control of an entity in the same role, and then one of the players relinquishes control by returning to the Choose Role Panel, the remaining player also loses control of that entity. If this happens, the remaining player can regain control of the entity by going through the Choose Role process. Note that two players can concurrently play different roles in the same vehicle, for example, one player is Driver and the other is Gunner or Commander.

3.2.1 Choosing an Existing Role

If you load a scenario or connect to a simulation that is publishing entities, you can select one of them as your role.

➤ **To choose a role:**

1. In the Choose Role Panel, select Control an Existing Entity. The panel lists the entities in the loaded scenario.
2. Optionally, filter the list by category, used by (country), force, or search string.
3. Optionally, switch from 3D view to the Plan View display mode (2D) by selecting an option at the bottom of the panel.
4. Select an entity by choosing it from the list or clicking it on the terrain.
5. If the selected entity has multiple roles, such as those on a vehicle, select an option from the Role list.
6. If the selected entity is an aircraft or a vehicle with different display options, select a display option from the Display(s) list. (For information about aircraft display options, see "7.1.4 Aircraft Display Options" on page 151.)
7. If you just want to observe the simulation from the perspective of the selected character but not actually control it, select the As Spectator check box.
8. If you want to use VR-Engage in full-screen mode, select the Full Screen check box.

   If you want to use VR-Engage in windowed mode, clear the Full Screen check box.
9. Click Engage. You can now control your character and participate in the scenario (or watch, if you chose to be a spectator).

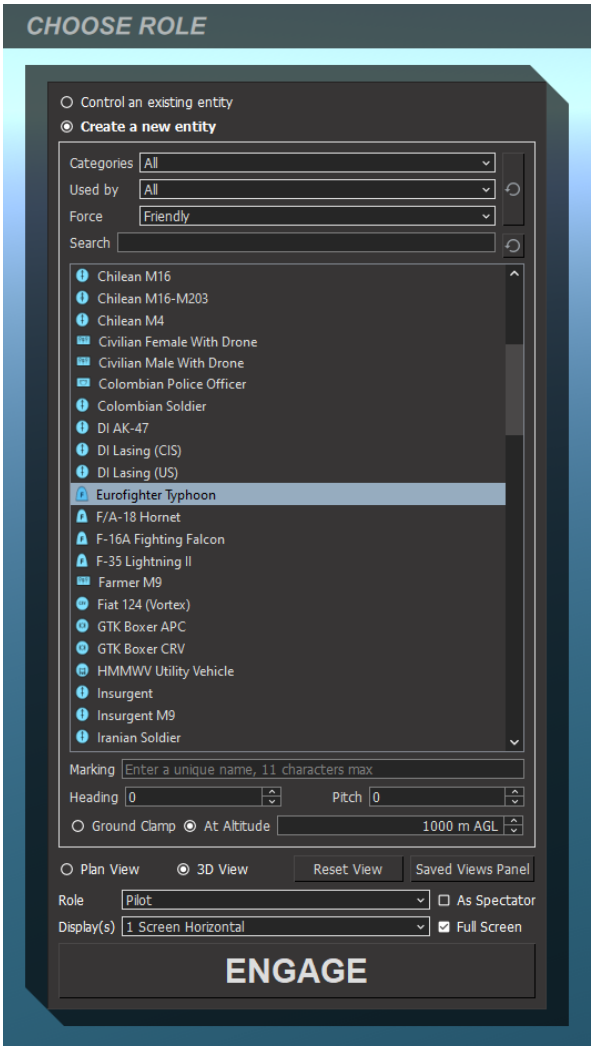
3.2.2 Creating a New Entity

When you are hosting a session or running an unhosted session, you can create a single entity and control it. That entity can participate in the simulation but, if you change roles, it is removed from the simulation.

➤ **To create an entity:**

1. In the Choose Role Panel, select the Create a New Entity option. The window lists the entity types that you can create.
2. Optionally, filter the list by category, country (used by), force, or search string.
3. Optionally, switch from the 3D view to the Plan View display mode (2D) by selecting an option at the bottom of the panel.
4. Select an entity in the list (Figure 11).

Figure 11. List of entities to create



5. Optionally, in the Marking box, you can type a name for the character. The maximum length is 11 characters.
6. Optionally, specify a heading.
7. Optionally, specify an altitude. By default, humans and ground vehicles are placed on the ground. Aircraft are placed at the altitude specified in the Altitude box. If you want an aircraft to be placed on the ground, you can change the specified altitude or select the Ground Clamp option. Similarly, if you want to create a paratrooper in the air, select the At Altitude option and specify an altitude.
8. Click on the terrain where you want to place the entity. (You can navigate the Plan View or 3D view using your mouse and keyboard. For details about navigating the terrain, see the chapter *Moving the Observer* in the *VR-Vantage Users Guide*.
9. If the selected entity has multiple roles, such as those on a vehicle, select an option from the Role list.
10. If the selected entity is an aircraft or a vehicle with multiple views, select a display option from the Display(s) list. (For information about aircraft display options, see "7.1.4 Aircraft Display Options" on page 151.)
11. If you simply want to observe the simulation from the perspective of the selected character but not actually control it, select the As Spectator check box.
12. If you want to use VR-Engage in full-screen mode, select the Full Screen check box.

    If you want to use VR-Engage in windowed mode, clear the Full Screen check box.
13. Click Engage. You can now control your character and participate in the scenario (assuming you did not choose to be a spectator).

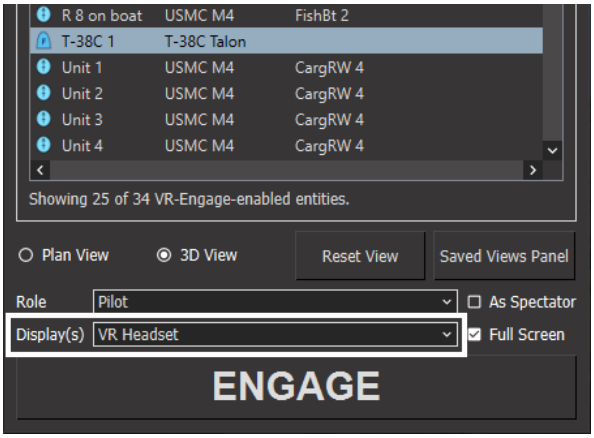
3.2.3 Using an HMD with an Entity

VR-Engage supports use of head-mounted displays to provide a virtual reality or mixed reality experience when you are controlling an entity. VR-Engage includes some entities that are configured for virtual or mixed reality. For entities that are configured for it, you can choose to use a VR or MR headset when one is connected. For details about configuring HMDs, see "3.10. Introduction to Head-Mounted Displays" on page 64.

➤ **To use a head-mounted display to control an entity:**

1. Start VR-Engage.
2. Join a session or load a scenario.
3. Create a new entity or select an existing entity.
4. On the Displays list, select VR Headset (virtual reality) or MR Headset (mixed reality), as appropriate.

Figure 12. Displays list



5. Engage the entity.

3.2.4 Assigning Roles from VR-Forces

If you install the VR-Engage Add-ons for VR-Forces in a compatible version of VR-Forces, you can assign roles from VR-Forces. This includes:

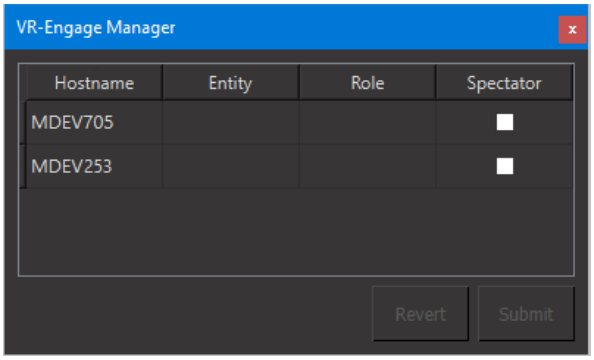
- Assigning an initial role for the player and starting the scenario.
- Specifying that the player be a spectator.
- Changing the role to None, which causes the VR-Engage session to revert to the Plan View role selection GUI.
- Changing the role to a different entity as the scenario progresses.

Managing entity assignments from VR-Forces does not prevent the VR-Engage player from using the VR-Engage GUI to select roles. If the VR-Engage player selects a role, that decision is reflected in the VR-Engage Manager Panel in VR-Forces.

➤ **To manage role assignments from VR-Forces:**

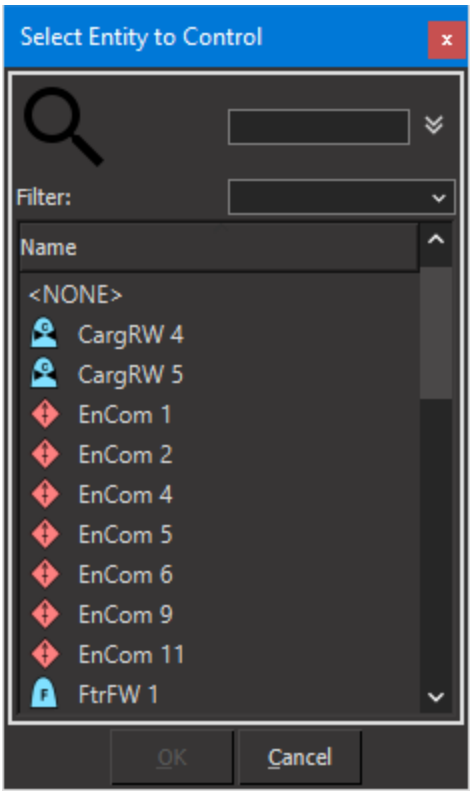
1. Start a version of VR-Forces that is compatible with your version of VR-Engage and has the VR-Engage Add-ons for VR-Forces installed.
2. Load the scenario you want to run.
3. Choose **View > VR-Engage Manager Panel**. The VR-Engage Manager Panel opens (Figure 13).

Figure 13. VR-Engage Manager Panel



4. Start VR-Engage. It automatically connects to the VR-Forces session. The VR-Engage Manager Panel displays the hostname of the machine running VR-Engage. (If you have multiple instances of VR-Engage running, they are listed on additional rows in the panel.)
5. Click the Entity cell for the VR-Engage host to which you want to assign a role. The Select Entity to Control dialog box opens (Figure 14).

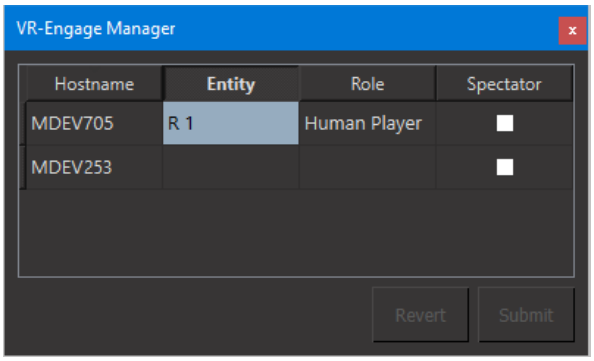
Figure 14. Select Entity to Control dialog box



6. Optionally, click the arrow to show the list. If you know the name of the entity, you can simply enter it.

7. If you're using the list, select the entity that you want the VR-Engage player to control. If the player has a role and you want to set it to no role, select <NONE>.
8. Click OK. The entity is listed in the table cell and the role is listed in the Role cell.

Figure 15. VR-Engage Manager Panel with entity selected



9. If the entity supports multiple roles, such as the LAV, you can select which role you want it to use. To select a role, click the Role cell. It becomes a list.
10. Select the role you want in the list. The panel automatically fills in the Station Name column.
11. If you want the player to participate as a spectator, select the Spectator check box.
12. Click Submit. The assignment is sent to the host. The player is automatically engaged in the assigned role on the host machine.

3.3. Connecting to VR-Forces

VR-Engage can interact with any DIS or HLA simulation. However, it has a particular ability to participate in simulations that use VR-Forces, MAK's computer generated forces application. This allows you to switch between player and CGF control while the scenario is running. You can also use VR-Forces to create scenarios that include VR-Engage controllable entities in addition to CGF simulation objects.

By default, when you start VR-Engage, it looks for a VR-Forces session to join. If it connects to a complementary VR-Forces build that has the VR-Engage plug-ins installed, you can take control of entities being simulated by VR-Forces. You can assume control of VR-Forces entities and give up control as desired.

When you take control of a VR-Forces entity, the entity pauses the plan or task it is executing. When you give up control of the entity, control returns to VR-Forces and the entity resumes whatever task or plan it was executing before you took control of it.

For a list of entities that can be controlled by VR-Engage players out of the box, see "Chapter 12. Controllable Entities" on page 239.

To learn how to create your own VR-Forces scenarios, see "Chapter 15. Creating VR-Engage Scenarios in VR-Forces" on page 297.



You can only control entities that are in the VR-Engage SMS or your own SMS that extends that SMS. The VR-Engage SMS includes most human characters, ground vehicles configured to use the Vortex libraries, and aircraft configured to use RotorLib or FixedWingLib.

3.4. Changing Your Role

If a scenario has multiple entities, you can change the role you are playing to be any entity in a scenario. You can change your role by exiting the simulation and choosing a new role. You can also change a role by embarking on a vehicle and choosing an available role.

3.4.1 Changing Your Role in the Choose Role Panel

You can change your role by returning to the Choose Role Panel. When you engage characters or exit a character to the Choose Role Panel, the state of the scenario may change. For details, see "3.5. Entity Engagement and Scenario State" on the next page.

➤ **To change your role:**

1. Press **Esc**. The Application Menu displays.
2. Click Choose Role. You return to the Choose Role Panel (Figure 10 on page 52).
3. Select the role you want, as described in "3.2. Choosing a Role" on page 52.
4. Click Engage.

3.4.2 Changing Your Role by Embarking on a Vehicle

You can change your role dynamically while you are participating in a simulation. The LAV has several roles associated with it: Driver, Gunner, Commander, Passenger, and Driver/Gunner. Other vehicles can have different sets of roles available. If you embark on a LAV while playing a human character, you can assume any of these roles except for Driver/Gunner. (If you embark on a LAV from the Choose Role panel, you cannot select Passenger.)

➤ **To change your role by embarking on a vehicle:**

1. Walk towards a vehicle that supports embarkation. When you get close to the vehicle, the Embark menu opens.

Figure 16. Embark LAV menu



If the Embark menu closes before you make a selection, back up your character a few steps and then approach the vehicle again.

2. Select a role using one of the following methods:
   - Use the up and down arrow keys or the up and down buttons on your D-pad to select a role. Then press the **Right arrow** key or right button on the D-pad to embark using the selected role.
   - Press the number key associated with the embarkation option you want.
   - Click the embarkation option you want. You can only use the mouse when mouse control is disabled.

3.5. Entity Engagement and Scenario State

The state of a scenario—running or paused—varies depending on whether or not you are actively engaged in controlling a character and which application started the scenario. The possible cases are as follows:

- If VR-Engage is hosting a session:
  - When a player clicks Engage to control an entity, it puts the scenario in running state. The word "Running" displays briefly at the top of the screen.
  - When a player presses **Esc**, the Application Menu displays. The player can click the pause button (") to pause the scenario. The word "Paused" displays at the top of the screen.

- If VR-Engage is joining a session hosted by VR-Forces or another instance of VR-Engage:
  - When a player clicks Engage to control an entity, it does not change the scenario's state.
  - When a player presses **Esc** to bring up the Application Menu, it does not change the scenario's state.
- If VR-Engage is starting an unhosted session:
  - When a player clicks Engage to control an entity, it does not change the scenario's state. The player must start the scenario.
  - When a player presses **Esc** to bring up the Application Menu, it does not change the scenario's state.

When the scenario is not running, the word "Paused" displays at the top of the screen. Players can only act while the scenario is running, not while it is paused.

If a player uses VR-Engage to join a session hosted by VR-Forces or another instance of VR-Engage and the user on the hosting application closes the scenario, a message informs the player that the session has ended and returns the player to the Searching for Session screen.

3.6. Overview of the Application Menu

When you are engaged in a role and press **Esc**, you see the Application Menu. This screen contains a number of options.

Figure 17. Application Menu



1. Click Return to Role to exit the Application Menu and return to the role you were playing.

2. Click Choose Role to return to the Choose Role panel.
3. Click Choose Session to leave the session and return to the Choose Session screen.
4. Click Exit VR-Engage to close the application.
5. The current status of the scenario, either Running or Paused.
6. Play the scenario.
7. Pause the scenario.
8. Rewind the scenario.

> The Play, Pause, and Rewind buttons are available only when you are hosting the session.

9. Click Back to exit a submenu, such as the Connection Menu. If you are not in a submenu, clicking Back exits the Application Menu and returns you to the role you were playing.
10. Display your connection information.
11. Open the control mappings.
12. Access the About dialog box. If you are in Admin mode, you can also access some advanced settings.
13. Close VR-Engage.
14. The status is information about your character and the scenario.

3.7. Starting a New Session

You can exit the session you are participating in and join or start a new session.

➤ **To change the scenario:**

1. Press **Esc**. The Application Menu displays.
2. Click Choose Session. Click OK in the Session Ended message.
3. Searching for Session screen displays.
4. Follow the procedure for starting a session, as described in "3.1. Starting VR-Engage" on page 49.

3.8. Configuring the Simulation Connection

To participate in a simulation, you must connect VR-Engage to a simulation network. VR-Engage includes connectors for DIS, HLA 1.3, HLA 1516, and HLA Evolved (IEEE 1516-2010, also referred to as HLA 1516e).

In User and Admin modes, you can change the configuration values of the default connectors and you can add connectors. You connect VR-Engage to a network at startup. To change the connection, click the Connection button at the top right.

For complete details about connection settings and parameters, see the *VR-Vantage Users Guide*.

➤ **To set up the simulation connection at startup:**

1. Start VR-Engage.
2. In the top left of the Connection Menu, select a connection.
3. At the bottom right of the Connection Menu, review the connection settings.

Figure 18. Connection Menu



4. Change the connection parameters as needed. If necessary, click the down arrows to access the advanced settings.
5. If you want to use those connection parameters in future sessions, click Save.
6. Click Connect to connect to the network.

3.9. Playing in Windowed Mode

By default, when you click Engage to control a character, you go into full-screen mode. There are no menus or other window controls available. If you want access to menus without escaping to the Choose Role Panel, you can exit full-screen mode and still control your character. This is called windowed mode.

Whenever you go from full-screen to windowed mode, VR-Engage releases mouse control so that you can use the mouse to access menus or other applications. You can also toggle mouse control on and off. (For more information about mouse control, see "5.1.1 Enabling and Disabling Mouse Control" on page 110.)



To access menus, you must be in Admin mode. See "3.1.3 Admin, User, and Lockdown Modes" on page 51.

➤ To change from full-screen mode to windowed mode, press **Ctrl**+**Enter**. To return to full-screen mode, press **Ctrl**+**Enter** again.

3.10. Introduction to Head-Mounted Displays

VR-Engage is integrated with several virtual reality head-mounted displays (HMDs). VR-Engage can render stereo views, which are presented inside the HMD. The head position and orientation tracking of the HMD are reflected in the observer as the wearer of the HMD moves their head.

VR-Engage uses the OpenVR, Oculus SDK, Varjo SDK, and VRgineers libraries to integrate with virtual reality systems. Any system that is capable of using OpenVR should work with VR-Engage.

VR-Engage is tested with the following systems:

- Oculus Rift and Rift S
- HTC Vive and HTC Vive Pro
- Valve Index
- Varjo XR-1, XR-3, VR-2, and VR-3
- VRgineers XTAL 2 and XTAL 3
- Windows Mixed Reality headsets

3.10.1 Prerequisites for Using an HMD

To use an HMD, you need specific software and VR-Engage setup:

- SteamVR. Required for the Vive and other OpenCR headsets. It is installed through the Steam digital distribution platform (*http://store.steampowered.com*).
- The Oculus Rift requires that you install the Oculus Runtime.
- For a Varjo headset, you must install Varjo Base. The Varjo XR-range devices support mixed reality, which requires that you also install Thunderbolt 3. For information on Varjo system requirements, see *https://varjo.com/use-center/getting-started/system-requirements*.
- XTAL headsets require the VRgineers software.
- If you are using Microsoft Mixed Reality devices, the "Windows Mixed Reality for SteamVR" add-on must be installed through Steam.



When using an HMD, ambient occlusion (HBAO+) is disabled by default.

3.10.2 Resetting the Tracking Sensor

For user comfort, you can reset the origin of the positional tracking performed by the HMD. This allows you to restart the head tracking once you have put on the HMD and assumed a comfortable position. You can reset the sensor at any time by pressing the **F8** key.

You can change the keyboard assignment by changing the “Reset Virtual Reality HMD” action in the VR-Engage Key Mapping Editor. (This action is available only when you run VR-Engage with the virtual reality plug-in.)



You can also reset the tracking position as follows:

- Choose **Reset VR Head Tracking** on the Action menu.
- Press the right thumbstick on a game controller.
- Press button 5 on the HOTAS Warthog joystick.

3.10.3 Calibrating the Eye Tracking for Varjo

Varjo devices allow applications that use eye tracking can start multi-dot calibration for more accurate eye tracking data. You can access the Varjo's eye tracking calibration by pressing **F9**.

3.10.4 Enabling and Disabling Virtual Reality Rendering

By default, when you start VR-Engage with an HMD connected, rendering to the HMD is enabled at startup. You can enable and disable rendering to the HMD at runtime.

You must be in Admin mode to manage the virtual reality settings. If you cannot see the menu bar, press **Shift**+**F12** to enter Admin mode.



If you are using a Vive HMD, you must power on the device before starting VR-Engage. Failure to do so leads to a situation in which HMD mode appears to be available as a display option in the Choose Role panel, but nothing happens if you select that display option and then engage an entity. Always ensure that the Vive is up and running before launching VR-Engage.

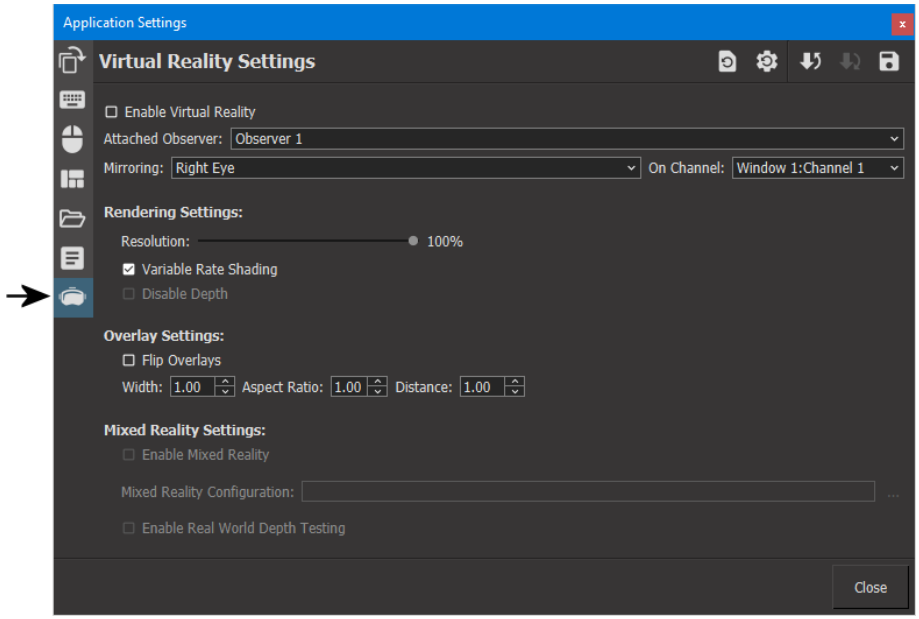
➤ **To enable or disable virtual reality rendering:**

1. Choose **Settings > Application**. The Application Settings dialog box opens.
2. Select the Virtual Reality Settings page (Figure 19).



The Virtual Reality Settings page is available only if the virtual reality plug-in is loaded and an HMD is detected.

Figure 19. Virtual Reality Settings page



3. Select or clear the Enable Virtual Reality check box.

> You can also enable or disable virtual reality by choosing **Settings > Enable Virtual Reality**.

3.10.5 Configuration Options for Virtual and Mixed Reality

You can set the following configuration options for virtual and mixed reality:

- Attached Observer. This is the observer that the HMD will attach to and render the perspective from. The default is to attach to Observer 1.
- Mirroring. This copies the image rendered for a given eye to the selected channel in the main display. The default is to render the right eye to the main channel. Mirroring disables the normal render pass of the attached channel to help with performance.

  > If mirroring is enabled and the HMD is attached to the observer that is driving a displayed channel, then the main view will be rendered from the HMD perspective. This affects performance, because the scene will be rendered a third time.

- Rendering Resolution. The slider bar allows you to dynamically adjust the resolution. Overlays remain at their native resolution.
- Variable Rate Shading. Improves performance in VR by lowering the precision and accuracy of GPU fragment operations in certain areas of the frame.

- Disable Depth. Some headsets support depth-based warping, which is a way of making image appear smooth with lower frame rates. Currently, this applies to Varjo headsets and the Oculus Rift. When you select Disable Depth, you disable the depth-based warp (VR-Engage does not pass the depth buffer to the headset SDKs). If the depth warp is causing cockpit windows to display artifacts and otherwise look incorrect, selecting Disable Depth can help.
- Flip Overlays. Currently this applies only to the HTC Vive. Enabling this option flips the overlays vertically. Select this check box when overlays display upside down.
- Enable Mixed Reality, MR Configuration, and Enable Real World Depth Testing. These options do not apply to VR-Engage.

➤ **To set virtual reality configuration settings:**

1. Choose **Settings > Application**. The Application Settings dialog box opens. (If the menu bar is not visible, press **Shift**+**F12** to enter Admin mode.)
2. Select the Virtual Reality Settings page (Figure 19 on the previous page).

   The Virtual Reality Settings page is available only when the virtual reality plug-in is loaded.

3. Change the settings as desired.

3.10.6 Setting Up Mixed Reality

VR-Engage supports the use of the forward-facing cameras on the Vive, Vive Pro, Varjo XR range, and XTAL headsets to add real elements such as a joystick, throttle, or other physical controls to the virtual world. Mixed reality modes are available for human roles and aircraft pilot roles for aircraft configured for VR. When you select a compatible entity, MR Headset is an option in the Display(s) list. Select this option to use mixed reality mode.

To use Mixed Reality mode in VR-Engage, you need a Vive, Vive Pro, Varjo XR, or XTAL, and the appropriate software (described in "3.10.1 Prerequisites for Using an HMD" on page 64). If you are using a Vive HMD, you also need a green screen and you must start SteamVR and turn on the HMD before you start VR-Engage. We strongly recommend that you test the Vive HMD camera before each use. For details, see "Testing the Vive Camera" on the next page.

For the Vive and Vive Pro, green screen masks out real-world elements that should not appear in the simulation. For the VR-Engage pilot role with mixed reality, the green screen is expected to cover an area approximately 120 degrees wide at -90 degrees elevation extending up to 60 degrees wide at 0 elevation, from the player's point of view. If seated at a table or desk, the desk should be covered with a green screen approximately five feet wide and extending up to at least seated eye level. The only controls that should be visible on top of the green screen are those that you want to see in the mixed reality view. The green screen should be well-illuminated with soft, diffused light so that the green color is clear and uniform.

Mixed Reality Masks

The MR_VR_Flight_Demo scenario contains examples of both VR and MR configurations in a T-38 fixed wing aircraft. VR-Engage allows you to specify the mask file used for a particular display layout. In this example, it's a mask of the lower part of the T-38's cockpit.

In the *./data/simulationModelSets/VR-Engage/roles/t38Pilot.lua* file, there is an MR Headset displayLayout. This demonstrates how to make a Mixed Reality displayLayout. To use a different mask for your own layout, refer to the "cameraProjectionGeometry" field. This is where the mixed reality mask definition file for the displayLayout is specified. For more information, see "Configuration for Mixed Reality" in the *VR-Vantage Users Guide*.

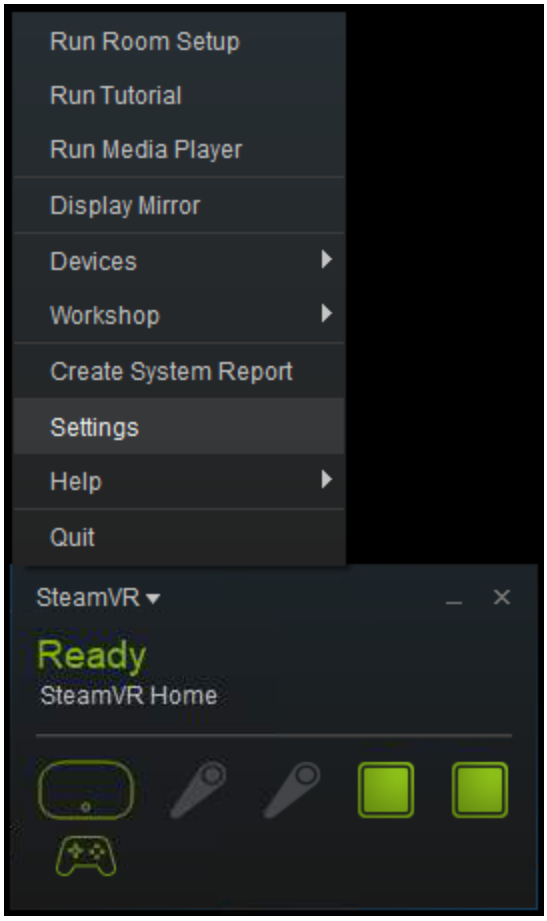
Testing the Vive Camera

If you are using the Vive HMD, you must turn it on before you start VR-Engage. We also recommend that you test the camera every time SteamVR or the computer is restarted, and prior to running VR-Engage. In some cases, the camera does not work unless you initialize it before starting VR-Engage.

➤ **To test the Vive camera:**

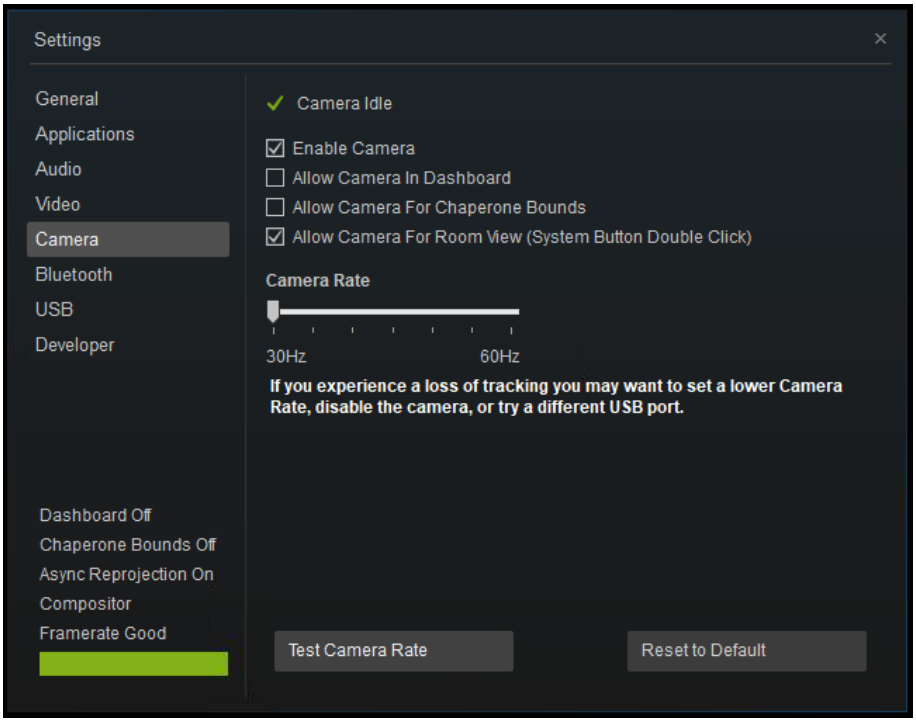
1. Turn on the Vive HMD.
2. Start SteamVR.
3. On the SteamVR menu, choose **Settings** (Figure 20). The Settings dialog box opens.

Figure 20. SteamVR menu



4. Select the Camera page (Figure 21).

Figure 21. Camera page



5. Click Test Camera Rate.
6. Confirm that the camera is working. (Wait until three green check marks appear.) If a red x appears, or if the Test Camera Rate button is not available, verify that the Vive is connected to a USB-3-capable port, and reboot the Vive headset and/or the computer.
7. If the test fails, restart the Vive HMD and SteamVR.

3.10.7 Optimizing Performance with HMDs

Virtual reality headsets require a high frame rate for a comfortable viewing experience. There are many factors that determine frame rate, such as complexity of the scene being rendered, terrain, special effects such as HDR and shadows, and environmental effects such as clouds. You may need to make trade-offs when selecting a terrain database, entity models, and applying certain visual effects at runtime.

For convenience, we have included an overlay installer that makes a number of these trade-offs to optimize performance in VR in the context air entities. The VR-Engage installation includes an installer named *MR_VR_FlightSettingsOverlay.exe*, located at the top level folder. Please see the associated *README_MR_VR_FlightSettingsOverlay.txt* for details on the settings it modifies.

3.11. Creating a Startup Configuration

You can create a startup configuration to automate some aspects of VR-Engage startup. VR-Engage includes a fully commented example startup configuration script called *exampleStartupConfig.lua* in *./appData/scripts*.

You can specify the following startup parameters in a script:

- **mode**. Tells VR-Engage to only `join` a running session, immediately `host` a session, or immediately `start` an unhosted session.
- **scenario**. Optional. If the **mode** is `host`, specifies the scenario to host. If you do not specify a scenario, VR-Engage uses the value for **terrain** or prompts you to specify a terrain.
- **terrain**. Specifies the terrain to load in `start` mode, or in `host` mode if you do not specify a scenario.
- **guiTerrain**. Optional. Specifies the terrain to use for visualization if you want to visualize on a terrain that is different from the simulation terrain.
- **attach**. Specifies an entity to control or attach to as a spectator.
- **entity**. Optional. If **attach** is not specified, this parameter specifies an entity to create.
- **minion**. Causes this instance of VR-Engage to connect to the entity being controlled by another instance of VR-Engage.

To use a startup configuration script, create a shortcut or batch file with the following command:

```
vrEngage.exe --initSettingsFile startupConfigFile
```

where:

- vrEngage.exe is the VR-Engage executable.
- *`startupConfigFile`* is the name of your startup configuration file, including the *.lua* extension. If you use a simple file name, VR-Engage looks for the file in *./appData/scripts*. You can also specify a relative path from the *./bin64* directory or a fully qualified path.

3.11.1 Minion and Boss Configuration

The **minion** startup configuration configures one station to watch the entity engagement status of another station, and automatically engage with that same entity. You would use it in conjunction with other startup options to create a multi-head station, where you have more than one instance of VR-Engage, on one or more machines, to create a full player interface.

The most common use case is for creating aircraft stations, although it applies to other entity types. You would have one or more VR-Engage stations running an out-the-window aircraft HUD display, and another one running the instrument panels.

On a single machine, a single instance of VR-Engage is able to create a separate instrument panel window, so this can be done with only one VR-Engage instance running. For performance reasons, however, you might want to use multiple machines. In that case, you need one instance of VR-Engage on one machine to run the out-of-window view, and a second instance of VR-Engage on another machine to run the instrument panel window. One of these is the Boss, which takes player input

and controls the entity, and the other is the Minion running as Spectator. The function of Minion mode then is to make these two stations behave as one when switching which entity is engaged.

The terms Master and Slave are technically a different context, which is why we use the terms Boss and Minion to differentiate.

3.12. Starting Multiple Simulation Engines

VR-Forces scenarios can be designed to use multiple simulation engines to improve performance. Each simulation engine simulates a subset of the simulation objects in the scenario. (This is called load balancing.) If a scenario is designed to use multiple simulation engines, VR-Engage can automatically start up the required number of simulation engines and benefit from load balancing.

By default, use of multiple simulation engines is disabled and the maximum number of simulation engines to use is set to 1. To enable use of multiple simulation engines, do one of the following:

- Start VR-Engage with the `--enableMultipleBackends` command-line argument.
- In *./appData/scripts/playerStation.lua*, uncomment the **enableMultipleBackends** parameter and set it to true.

By default, the maximum number of simulation engines that VR-Engage can start is `number of logical processors/2`. The minimum is always 1. You can change the maximum number of simulation engines to start by uncommenting the **MaxBackends** parameter in *./appData/scripts/playerStation.lua* and setting it to your desired value.

If you enable multiple simulation engines but do not have enough licenses for the number of simulation engines that a scenario is configured to use, VR-Engage simulates all entities on one simulation engine.

The VR-Engage_First_Experience scenario is configured to use two simulation engines. The MAK_Suite_Demo_California scenario is configured to use three simulation engines.

If you start one of these scenarios in VR-Forces and do not have the configured number of simulation engines running, VR-Forces prompts you to remap the entities to the existing simulation engines. For information about load balancing and remapping scenarios, see the *VR-Forces Users Guide*.

You must have a VR-Forces license for each simulation engine you start.

3.13. Command-Line Options

Table 2 lists VR-Engage's command-line options.

VR-Engage also allows you to use startup and player station configuration files to determine some options when the program launches. See "3.11. Creating a Startup Configuration" on page 70 and "3.14. Command-Line Options for Player Stations" on page 85.

Table 2. VR-Engage command-line options

Option	Description
(-- | --ignore_rest)	Ignore all command-line options after this one. This option is useful for disabling part of a command line in a script for test purposes so that you do not have to retype all of the command options at a later point.
--airMode	Launches VR-Engage in Air mode. You can create and control only aircraft and humans.
--antiAliasing *level*	Set the anti-aliasing level (0, 2, 4, 8, or 16). Default: 4. Depending on your graphics card and driver, some combinations of anti-aliasing and bit settings (--depthBits, --stencilBits, and --alphaBits) may be incompatible. For example, if you set stencil bits to 8, the depth must be set to 24 and anti-aliasing may need to be set to 0 or 2.
--appCacheDir *directory*	Use --appCacheDir to specify a directory for the cache. This allows you to run multiple remote display engines (RDEs) or other instances of VR-Engage from the same installation. Running two RDEs on the same machine helps compensate for the limited threading in VR-Engage.
--appDataDir *directory*	Specifies the location of application data.
--appMode [user|admin|lockdown]	Controls the features that are available to the user. • User mode does not display the menus when you start VR-Engage. • Admin mode shows the menus when you start VR-Engage. User mode is the default behavior. In VR-Engage, you can toggle between the User and Admin modes by pressing Shift+F12.
--autoReconnectPeriod *period*	Specifies the amount of time, in seconds, to wait before trying to attach to disconnected remote displays.

Table 2. VR-Engage command-line options (continued)

Option	Description
(-B | --bindObservers) *"observer1,..., observern,../saved_views_file" "../terrain_file"*	If you load a scene that has multiple observers, this command assigns the views in the specified saved views file to the specified observers. The first saved view is assigned to the first observer, the second to the second observer, and so on. If the observers are not specified as arguments, only the first observer in the display engine configuration gets a view assigned to it. The entire string must be quoted. There are no spaces after commas. You must specify a terrain database for this command to work. You would ordinarily also specify a display configuration that contains the observers with --dispSetting. For more information about using this command-line option, see "Attaching the Observer at Startup" in the VR-Vantage Users Guide, the attaching the observer to an entity at startup tutorial.
(-c | --showConsole)	Display a console window.
(-C | --autoConnect) *{filepath | connector_name}*	Automatically connect to the network using the specified connector.
--dataDir *directory*	Specifies the location of the data directory (terrains and so on).
--depthBits *bits*	Sets the depth of graphics to 24 bits or 32 bits. This has no effect when reverse Z is enabled, which is the default for VR-Engage. The number of depth bits in reverse Z is always 32 bits float.
--disableBufferClearUsesCompute	This is a developer option for debugging that should not be included in customer-facing documentation (VRV-2672). It's just here for when we check command-line options lists. Disables the use of compute shaders for clearing buffers.
--disableDynamicTerrain	Disables the dynamic terrain engine. May slightly speed up terrain loading and paging.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--disableErrorDialog`	Prevents error messages from displaying. Errors print to the console instead.
`--disableEscapeKey`	Disable use of the **Esc** key while the player is engaged in a role.
`--disableIndirectEntities`	Disables indirect rendering of entities. To enable indirect rendering, use `--enableIndirectEntities`.
`--disableOcclusionCulling`	Disables occlusion culling. This may improve performance.
`(-z | --disableReverseZBuffer)`	Disables use of reverse Z buffering, which is enabled by default. Leaving reverse Z buffering enabled can reduce Z-fighting when viewing the terrain from a distance. (3D observer modes only.)
`--disableTerrainObserverViews`	Disables the loading of terrain observer view files. Loading a terrain does not clear the observer views and the terrain's observer views do not load.
`--discoveredModelPreloadFileName` *filename*	Specifies a file to which the list of models discovered during a session that were not preloaded is written.
`--displayEnginePort` *port*	Specifies the display engine port used by a remote display engine. Used only if `--slave` is specified.
`--dispSetting` *filename*	Specifies the display engine configuration to load.
`(-e | --extraArgs)` *string* | "*string1 ... stringn*"	Allows you to send arbitrary strings to the initializer, which can be queried for by plug-ins. This option works around the restriction on adding new command-line arguments. It can take one argument or multiple arguments. Multiple arguments must be wrapped in double quotes.
`(-E | --entTypeMap)` *filename*	Load the specified object type mapping file. The file extension determines the type of mapping.
`--enableDofConstraints`	Enables the use of degree(s) of freedom constraints on simulation object models.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--enableMultipleBackends`	Causes VR-Engage to load the number of back-ends required by a scenario up to the maximum number specified in *./appData/scripts/playerStation.lua*.
`--enableSensorFX`	Load the SensorFX plug-in. SensorFX uses more advanced rendering to display infrared and night vision.
`--enableTracyProfiling`	Enables the Tracy Client when you launch the application.
`--enableVSync`	Turns on VSync.
`--entDispSetting` *filename*	Specifies the entity display setting to load.
`--entityGridCellSize` *x y z* `--entityGridCellSize` *x,y,z* `--entityGridCellSize` *integer*	Use this option for debugging, particularly looking for bottlenecks in the code. Entities (in indirect) are divided into a grid of cells. The cells have the same size. This is a regular (sparse) grid that allows for efficient, higher-level culling. The default cell size is 5000 5000 5000. You can specify three values or one, for example: • 2000 2000 2000 • 2000 • 2000 200 10000 • -1 -1 -1 (one cell, infinite cell size) • -1 (one cell, infinite cell size)
`--entityDefsDir` *directory*	Loads the *.hier* and *.leaf* files in the specified directory.
`--envSetting` *filename*	Specifies the environment configuration file to load.
`--factoryRootDir` *directory*	Set the root directory used to restore settings to factory defaults.
`--forceTextShaping`	Force all text to use HarfBuzz shaping. Use this option if you need support for right-to-left languages. It may affect performance.
(`-F` | `--frameRate`) *rate*	If rate is 0, the frame rate is variable. Otherwise, specifies a fixed frame rate for rendering.
`--fullScreen`	Start with the window in full screen mode.
(`-G` | `--locale`) *language*	Specifies the locale (for localization).

Table 2. VR-Engage command-line options (continued)

Option	Description
`--groundMode`	Run VR-Engage in Ground mode. You can create and control only ground vehicles and humans.
`--guiWindowRectangle "`*`width height x y`*`"` `--guiWindowRectangle` *`width`*`,`*`height`*`,`*`x`*`,`*`y`*	Specifies the window size and location at startup.
`(-h | --help)`	Display command-line usage information.
`--highestPriority`	Changes the priority of the application and the main render thread to the highest available priority. This command may starve other processes.
`--initSettingsFile` *`filename`*	Name of the initialization settings file to load.
`(-K | --disableKDTrees)`	Disable creation of KD trees for intersections.
`(-l | --logFileName)` *`filename`*	Specifies the log file name.

`(-L | --loadObserverViews)` *`filename`*	Loads the saved observer view states in the specified file. You can use this command multiple times, for example:
	`--loadObserverViews file1`
	`--loadObserverViews file2`

`--limitEntityTextures "`*`width height`*`"`	Uses only mipmap levels at or below the resolution for entities. Use two integers for the maximum width and height of the texture mipmap to use, for example, `1024 1024` or `2k 2k`. To turn off the texture limit function, use `--limitEntityTextures "-1 -1"`.
`--limitTerrainTextures "`*`width height`*`"`	Uses only mipmap levels at or below the resolution for terrain. Use two integers for the width and height, for example, `1024 1024` or `2k 2k`.

In most situations, using `--limitEntityTextures` and `--limitTerrainTextures` prevents crashes with high VRAM. If you continue to experience crashes, use `--limitGpuMemoryPercentage` in addition to these command-line options.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--limitGpuMemoryPercentage` *integer*	Set the cutoff percentage as a value between 0 and 100. (Typical values are between 85 and 100.) Beyond the specified percentage, VR-Engage does not make new textures or buffers as a percentage of the available video RAM. This applies only to VR-Engage-side textures: terrain insets, entities, indirected osgEarth layers (feature_geom and buildings). It does not apply to SilverLining, Triton, or osgEarth tiles. The latter are controlled by the *.earth* file options.
 Typically, you would use `--limitGpuMemoryPercentage` in conjunction with `--limitEntityTextures` and `--limitTerrainTextures` when you see crashes with high VRAM.	
`--mainScreenNum` *num*	Specifies the display that the main window should use. The screen number can be a number from 0 to the number of screens -1. This number matches the screen number used in the Display Engine Configuration Editor for a window.
`--mergeEntityTypeMap {`*directory* | *filename*`}`	Merges all entity type mapping files (**.metx*) in a directory or individual entity type mapping files. Can be used multiple times.
`--mergeHierarchyEntityDef {`*directory* | *filename*`}`	Merges the *.hier* files in the specified directory or the individual *.hier* file into the default hierarchy (usually loaded from *./SharedData/16 /latest/ModelData/Configuration/Original/Definitions*.) Merging is additive only. No nodes in a previously loaded hierarchy are replaced. Can be used multiple times.
`--mergeHierarchyTacticalGraphicsDef {`*directory* | *filename*`}`	Merges the specified *.hier* file(s) into the default hierarchy (usually loaded from *./SharedData/16 /latest/ModelData/Configuration/Original/Definitions*.) Merging is additive only. No nodes in a previously loaded hierarchy are replaced. Can be used multiple times.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--mergeHierarchyUnitDef {`*`directory`* `|` *`filename`*`}`	Merges the specified *.hier* file(s) into the default hierarchy (usually loaded from *./SharedData/16 /latest/ModelData/Configuration/Original/Definitions*.) Merging is additive only. No nodes in a previously loaded hierarchy are replaced. Can be used multiple times.
`--mergeLeafEntityDef {`*`directory`* `|` *`filename`*`}`	Merges the leaf files in the specified directory or the individual leaf file into the default entity element definitions. Can be used multiple times.
`--mergeLeafTacticalGraphicsDef {`*`directory`* `|` *`filename`*`}`	Merges the leaf files in the specified directory or the individual leaf file into the default tactical graphics element definitions. Can be used multiple times.
`--mergeLeafUnitDef {`*`directory`* `|` *`filename`*`}`	Merges the leaf files in the directory or the specified individual leaf file into the default unit element definitions. Can be used multiple times.
`--mergeModelDefs {`*`directory`* `|` *`filename`*`}`	Merges all model definition files (**.ommx*) in the specified directory or individual model definition files. Can be used multiple times.
`--mergeTacticalGraphicsTypeMap {`*`directory`* `|` *`filename`*`}`	Merges all tactical graphics type mapping files (**.mtgx*) in a directory or individual tactical graphics type mapping files. Can be used multiple times.
`--mergeUnitTypeMap {`*`directory`* `|` *`filename`*`}`	Merges all unit type mapping files (**.magx*) in a directory or individual unit type mapping files. Can be used multiple times.
`--modelPreloadFileName` *`filename`*	Specifies a file that has a list of models to preload.
`(-n | --notifyLevel)` *`notify_level`*	Specifies the notification level for application messages, where *notify_level* is a number from 0 through 4. The least verbose message level is 0, the highest is 4. The default is 2.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--noAppDataEntityDefs`	Do not load the entity element definitions in *./SharedData/16/latest/ModelData/Configuration/Definitions*.
`--noAppDataEntityTypeMap`	Do not load entity type mappings in *./SharedData/16/latest/ModelData/Configuration/Definitions*.
`--noAppDataModelDefs`	Do not load model definitions in *./SharedData/16/latest/ModelData/Configuration/Definitions*.
`--noAppDataTacticalGraphicsDefs`	Do not load the tactical graphics element definitions in *./SharedData/16/latest/ModelData/Configuration/Definitions*.
`--noAppDataTacticalGraphicsTypeMap`	Do not load tactical graphics type mappings in *./SharedData/16/latest/ModelData/Configuration/Definitions*.
`--noAppDataUnitDefs`	Do not load the unit element definitions in *./SharedData/16/latest/ModelData/Configuration/Definitions*.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--noAppDataUnitTypeMap`	Do not load unit type mappings in *./SharedData/* *16* */latest/ModelData/Configuration/Definitions* .
`--noGui`	Start without the graphical user interface.
`(-O | --hasNTPSync)`	Enables the synchronization of all time-based features between different VR-Vantage applications, such as VR-Vantage and the VR-Forces front-end. System clocks must be synchronized. Time-based features include the rendering of ocean waves, clouds, and SpeedTrees.
`--plugin` *filename*	Load the specified plug-in. Can be used multiple times.
`--pseudoFullScreen`	Use pseudo full screen for full screen mode. This may improve performance when using dynamic ocean.
`-m, --realizeMinimap`	Realize the minimap model set.
`(-s |--slave)`	Run as a remote display engine.
`(-S | --suppressHatIntersectOnAttach)`	Suppress the periodic HAT intersection when attached. This setting can improve performance if you need to attach the observer to ground vehicles a lot, but do not need to attach to air vehicles.
`--sensorMode`	Run VR-Engage Sensor Operator. You can create and control only sensors and humans.
`--sharedDataDir` *directory path*	Defines the location of the shared data directory if you installed it somewhere other than the default.
`--sharedDataRoot` *directory root*	Defines the location of the shared data root.
`--sharedDataStem` *directorystem*	Defines the location of the shared data stem.
`--stationName`	Station name of application as seen by the VR-Engage manager.
`--stencilBits` *bits*	Sets the number of stencil bits (0 or 8). Default: 0.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--tcpSendTimeout`	Sets the timeout (in seconds) for the master display engine to wait for a remote display engine before giving up on a send. Default: 30 seconds.
(*terrain_MTF_file* | *scene_MSF_file*) *terrain_MTF_file*	Specifies a terrain (MTF) or scene (MSF) file to load. The path must be a relative path (from *./bin64*) to the file, for example, *../userData/terrains/MTF_file*.
`--unplug` *filename*	Do not load the specified plug-in. Can be used multiple times.
`--userDataDir` *directory*	Specifies the user data directory.
(`-v` | `--version`)	Display version information and exit.
Windows syntax: `-W | --withDisplayEngine` *port@netAddress;displayConfigFilename1 .... n* Linux syntax: `-W | --withDisplayEngine` *port@netAddress:displayConfigFilename1 .... n*	Connect to a display engine running on another computer. For more information, see "Loading Multiple Display Engine Configurations from the Command Line" under "Loading a Display Engine Configuration" in *VR-Vantage Users Guide*.
(`-Z` | `--ignoreZoomForSpeedtreeLod`)	Disable zoom for SpeedTree LOD.
HLA Only	**These options are available if you specify --hla13, --hla1516, or --hla1516e.**
(`-a` | `--appNumber`) *number*	Application number.
(`-f` | `--fomMapperLib`) *libname*	FOM Mapper library name.
(`-F` | `--fedFileName`) *fedfile_name*	FED file name.
`--fileTransporterReceivePort` *port*	Port for the file transporter. This is the port used to receive order of battle and plan files.
`--fomMapperInitData` *data*	FOM Mapper initialization data.
`--fomModules` *module*	Specifies an HLA Evolved FOM module. Can be used multiple times.
(`-H` | `--hostAddressString`) *address*	Specifies the host address. This is usually the same as the device address.
(`-i` | `--sessionId`) *ID*	Specifies the session ID for this instance of the front-end.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--ignoreAdvisories`	Enables or disables HLA advisory messages. Default: true.
`--localSettingsDesignator` *designator*	Specifies a string that is passed to the RTI during federate connect when using HLA Evolved. It is usually used to provide additional RTI configuration, but every RTI uses it differently. Please consult your RTI documentation for proper usage.
(`-N` | `--federateName`) *federate_name*	Specifies a unique name that can be assigned to the federate. No two federates within the same federation can have the same name. If no name is specified, the RTI will automatically assign a unique name for the federate. Default: VR-Forces. (HLA Evolved only)
(`-p` | `--federateType`) *type*	Federate Type can be used to distinguish different categories of federates, and its only practical use is in save-and-restore.
`--fileTransporterReceivePort` *port*	Port for the file transporter. This is the port used to receive order of battle and plan files.
`--reportedHostAddressString` *address*	Specifies how VR-Forces reports its IP address to another VR-Forces application for the purpose of file transfers. It is only important when you have a GUI or remote controller on one side of a router and a simulation engine on the other, and there is some kind of network address translation occurring whereby they have to connect to one another using an IP address other than the one they recognize themselves at locally.
`--rprFomRevision` *revision_number*	RPR FOM revision for RPR FOM 2.0, draft 17.
`--rprFomVersion` *version_number*	RPR FOM version. (0.5, 0.7, 0.8, 1.0, 2.0006, 2.0014, 2.0017, or 2.0)
(`-s` | `--siteId` *ID*)	Site ID.
`--useAbsoluteTimeStamps`	Enables use of absolute time stamps. Otherwise uses relative time stamps.

Table 2. VR-Engage command-line options (continued)

Option	Description
If you select **--useAbsoluteTimeStamps**, it is your responsibility to ensure that the time clocks of exercise participants are synchronized. If they are not synchronized,VR-Engage may not work properly.	
`--useGeographicDdm`	Specifies use of geographic DDM if you are using the MAK RTI. Default: false.
`(-x | --execName)` *exec_name*	Execution name. Default: MAK-RPR-2.0.
DIS Only	**These options are available if you specify --dis.**
`(-a | --appNumber)` *number*	Application number.
`(-A | --destAddrString)` *address*	Destination address.
`--appIdRange` *range*	The application ID range for VR-Forces back-ends.
`--defaultObjectTimeout` *timeout*	Specifies the timeout interval for objects.
`--deviceAddress` *address*	Specifies the address of the network card to use for UDP traffic.
`--disVersion`	The DIS version. Default: 7.
`--fileTransporterReceivePort` *port*	Port for the file transporter. This is the port used to receive order of battle and plan files.
`(-H | --hostAddressString)` *address*	Specifies the host address. This is usually the same as the device address.
`--mcastTtl` *time*	Specifies the multicast time-to-live. Default: -1.
`--multicastAddresses` *address1 ... addressn*	Specifies one or more multicast addresses.
`(-P | --disPort)` *port*	DIS port.
`--recvBufferSize` *size*	Receive buffer size.
`--reportedHostAddressString` *address*	Specifies how VR-Forces reports its IP address to another VR-Forces application for the purpose of file transfers. It is only important when you have a GUI or remote controller on one side of a router and a simulation engine on the other, and there is some kind of network address translation occurring whereby they have to connect to one another using an IP address other than the one they recognize themselves at locally.

Table 2. VR-Engage command-line options (continued)

Option	Description
`--useAbsoluteTimeStamps`	Enables use of absolute time stamps. Otherwise uses relative time stamps.
⚠ If you select **--useAbsoluteTimeStamps**, it is your responsibility to ensure that the time clocks of exercise participants are synchronized. If they are not synchronized,VR-Engage may not work properly.	
`(-s | --siteId` *`ID`*`)`	Site ID.
`--sendBufferSize` *`size`*	Send buffer size. Default: -1.
`--subnetMask` *`mask`*	Specifies a subnet mask.
`--useAsyncIO`	Use asynchronous I/O. Default: False.
`--useIpv6`	Use the IPV6 protocol, if available.
`(-x | --exerciseId)` *`ID`*	Exercise ID.

3.14. Command-Line Options for Player Stations

You can specify additional command-line options by editing *./appData/scripts/playerStation.lua* and adding command-line options to **additionalBackendCommandLineParameters**. For example:

```
additionalBackendCommandLineParameters = --outputLogFile my-log-file
--startInRunMode --userDataDir my-data-directory
```

Please note:

- The connection settings in the VR-Engage GUI's current connection options. A warning displays in the GUI console and in *vrEngage.log* for each ignored parameter, and what the replacement is.
- If you supply a DIS-specific command to HLA, or vice versa, the application will not start correctly.

3.15. Managing Plug-ins

One way to extend the functionality of VR-Engage is by loading plug-ins that have been written for it by MAK or other software vendors. You can manage which plug-ins get loaded when VR-Engage starts up. You can also view information about the plug-ins that have been loaded. This is primarily used for back-end plug-ins.

Plug-ins are implemented as one or more dynamic linked libraries (DLL or SO files). The DLLs can be release or debug, for vrfSim or vrfGui, for Windows or Linux, and for a specific protocol or protocol-independent. Each association of an application, platform, build type, and transport type is called a configuration. The Plug-ins Editor lets you specify the DLLs to use for each of the desired configurations.

- Changes to plug-in configurations do not take effect until the next time you start the affected VR-Engage application.
- To access the Settings menu, you must be in Admin mode. See "3.1.3 Admin, User, and Lockdown Modes" on page 51.

3.15.1 Loading Plug-ins

You can specify which plug-ins to load in either of the following ways: using the Plug-ins dialog box. Those plug-ins load the next time VR-Engage starts.

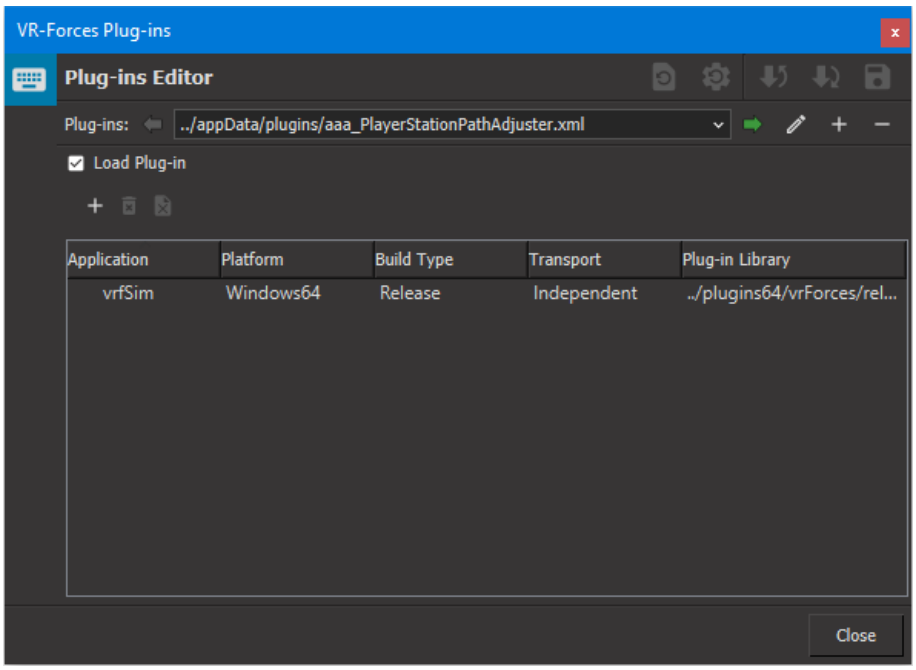
Specifying the Plug-ins to Load in the Plug-ins Dialog Box

The Plug-ins dialog box lets you specify the plug-ins that will be loaded the next time that you start VR-Engage.

➤ **To specify that a plug-in be loaded or not loaded in the Plug-ins dialog box:**

1. Choose **Settings > Plug-ins.** The Plug-ins dialog box opens (Figure 22).

Figure 22. Plug-ins Editor



2. In the list of plug-ins at the top of the page, select the one that you want to edit. The configurations for that plug-in are listed in the window.
3. Select or clear the Load Plug-in check box.

3.15.2 Adding a Plug-in

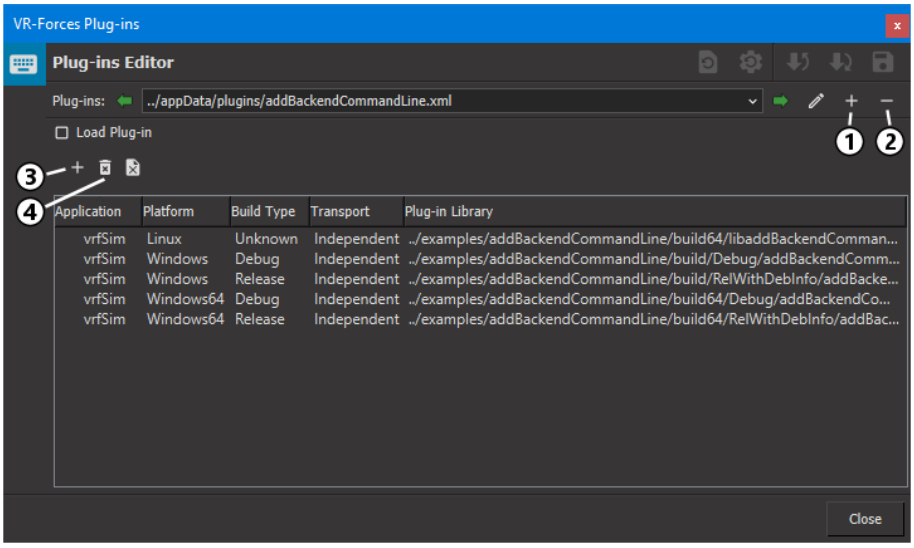
When you add a plug-in, VR-Engage creates an XML file (in *./appData/plugins*) that contains all of the information VR-Engage needs to load the plug-in. New plug-ins are added with entries for each of the configurations (application, platform, build type,

transport). After you add a plug-in, you must specify the DLLs for each configuration that you want to support.

➤ **To add a plug-in:**

1. Choose **Settings > Plug-ins**. The Plug-ins dialog box opens (Figure 22 on the previous page).
2. Select the Plug-ins Editor page.
3. Click the Add button (**+**) at the upper right corner of the page (#1 in Figure 23). The New Plug-in dialog box opens.
4. Type a name for the plug-in.
5. Click OK. The plug-in is added to the Plug-ins list. It has configurations for each of the combinations of application, platform, build type, and transport (Figure 23).

Figure 23. New plug-in



6. Specify the DLL for each configuration that you want to support, as described in "3.15.3 Specifying the DLLs for a Plug-in" below.

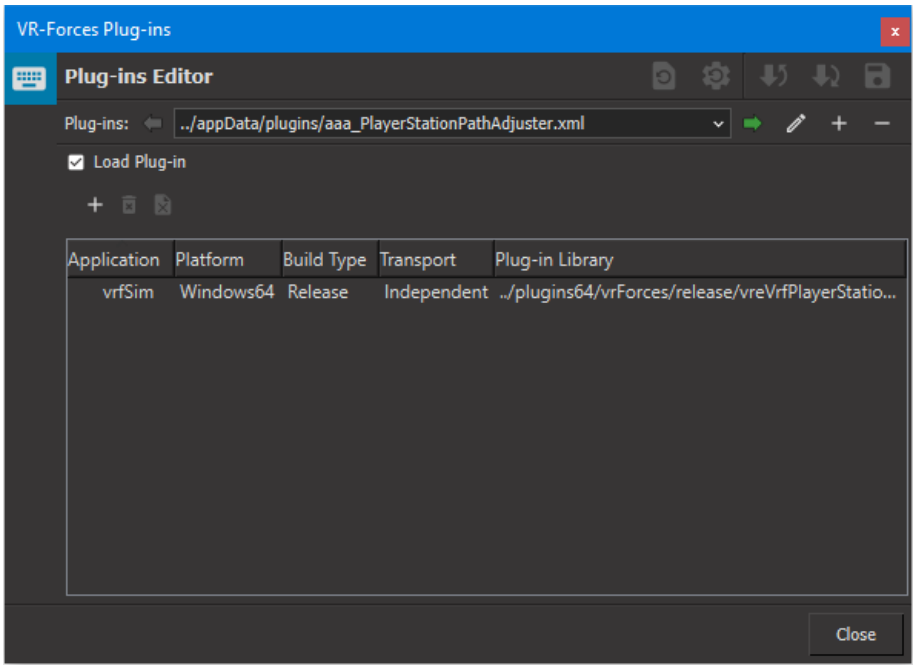
3.15.3 Specifying the DLLs for a Plug-in

Each plug-in configuration that you want to be able to load must specify the DLL to load for that configuration.

➤ **To specify the DLL for a plug-in configuration:**

1. Choose **Settings > Plug-ins.** The Plug-ins dialog box opens (Figure 22 on page 86).

Figure 24. Plug-ins Editor



2. In the list of plug-ins at the top of the page, select the one that you want to edit.
3. Click the Plug-in Library column for the configuration that you want to edit. The Select Plugin dialog box opens.
4. Select the DLL that you want to load for this configuration.
5. Click Open. The value is changed.

Removing a Plug-in's DLL

You can remove the DLL assignment for a plug-in configuration without removing the configuration.

➤ **To remove the DLL assignment for a plug-in:**

1. Choose **Settings > Plug-ins.** The Plug-ins dialog box opens (Figure 22 on page 86).
2. Select the configuration whose DLL assignment you want to remove.
3. Click the Remove button .

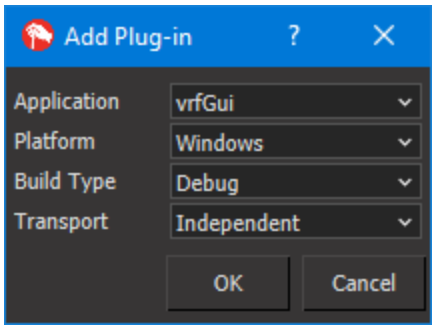
3.15.4 Adding a Plug-in Configuration

A plug-in configuration is a row in the Plug-ins Editor window. It specifies an application, platform, build type, transport type, and plug-in library.

➤ **To add a plug-in configuration:**

1. Choose **Settings > Plug-ins**. The Plug-ins dialog box opens (Figure 22 on page 86).
2. In the list of plug-ins at the top of the page, select the one that you want to edit.
3. Click the Add button (+) above the list of configurations (#3 in Figure 23 on page 87). The Add Plug-in dialog box opens (Figure 25).

Figure 25. Add Plug-in dialog box



4. For each parameter (Application, Platform, Build Type, and Transport), select a value from the list.
5. Click OK. The configuration is added to the list.
6. Specify a DLL for the configuration, as described in "3.15.3 Specifying the DLLs for a Plug-in" on page 87.

3.15.5 Deleting a Plug-in Configuration

A plug-in configuration is a row in the Plug-ins Editor window. It specifies an application, platform, build type, transport type, and plug-in library.

➤ **To delete a plug-in configuration:**

1. Choose **Settings > Plug-ins**. The Plug-ins dialog box opens (Figure 22 on page 86).
2. In the list of plug-ins at the top of the page, select the one that you want to edit.
3. Select the configuration that you want to delete.
4. Click the Delete button ( ) that is above the list of configurations (#4 in Figure 23 on page 87).

3.15.6 Deleting a Plug-in

When you delete a plug-in, you delete the XML configuration file in *./appData/plugins* that specifies the plug-in configurations and whether or not to load it.

➤ **To delete a plug-in:**

1. Choose **Settings > Plug-ins**. The Plug-ins dialog box opens (Figure 22 on page 86).
2. In the list of plug-ins at the top of the page, select the one that you want to delete.

3. Click the Delete button (−) to the right of the plug-ins list (#2 in Figure 23 on page 87).



4. Common Player and Entity Features

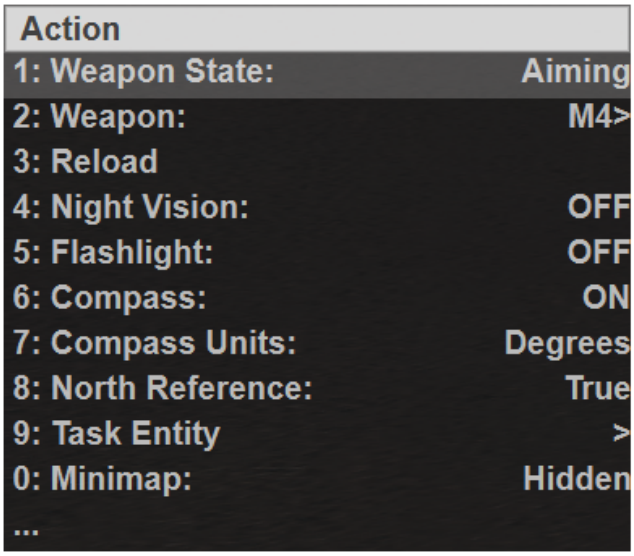
This chapter describes actions that are common to all types of entities in a scenario.

4.1. The Action Menu 92
4.1.1 Selecting Options on the Action Menu 92
4.2. Using the Radio 93
4.2.1 Using MAK Radio in VR-Forces 94
4.2.2 Using the Standalone Radio Application 94
4.2.3 Using the Intercom or a VOX Radio 96
4.2.4 Configuring the MAK Radio 96
4.2.5 Audio Settings 100
4.2.6 Jamming Radios 101
4.3. Changing the View Mode 101
4.4. Starting VR-Forces Events 102
4.5. Introduction to Sound Effects 102
4.5.1 Enabling Sound Effects 103
4.5.2 Configuring Sound Effects 103
4.5.3 Decibels Configuration 104
4.5.4 Associating Sound Files with Entities and Events 105
4.6. Resetting Entities 105
4.7. HMD Controls 106
4.8. Saved Observer Views 106
4.8.1 Selecting a View 107
4.8.2 Transitioning to a View 107
4.8.3 Resetting the View 107

4.1. The Action Menu

The Action menu lets you turn several features on and off (toggles) and lets you assign tasks to other entities if you are hosting a VR-Forces scenario or connected to a VR-Forces session. The Action menu is context-sensitive.

Figure 26. Action menu for desktop



➤ **To open the Action menu, do either of the following:**

- Press ~.
- Press the menu button on your game controller or head-mounted display (HMD) controller (the Back button on game controllers).

The Oculus Rift, HTC Vive, and Varjo HMDs configure the Action menu differently. If the Action menu is displayed upside down, you must change the configuration. For details, see "3.10.5 Configuration Options for Virtual and Mixed Reality" on page 66.

4.1.1 Selecting Options on the Action Menu

You can choose options on the Action menu using the keyboard (arrow keys or number keys), the D-pad on a game controller or HMD controller, or your mouse.

➤ **To choose an option on the Action menu using the arrow keys or D-pad:**

1. Press the **Up arrow** and **Down arrow** keys on your keyboard or the up and down buttons on a game controller's D-pad to highlight an option.
2. Press the **Right arrow** key or the right button on the D-pad to toggle the option on or off or to display a submenu.

When you choose an option with the mouse, if it is a toggle, the state changes immediately. If it displays a submenu, the menu changes immediately. If it is a task, the entity executes that task immediately. Therefore, if a task requires input, such as a location, do not use the mouse to choose it. Use the arrow keys.

➤ To choose an option on the Action menu using a mouse, click the menu option.

Each option on the Action menu has a number. When you choose an option with a number key, if it is a toggle, the state changes immediately. If it displays a submenu, the menu changes immediately. If it is a task, the entity executes that task immediately. Therefore, if a task requires input, such as a location, do not use a number key to choose it. Use the arrow keys.



Numbers are reassigned dynamically if you scroll the Action menu or menu commands are removed due to some change in the state of the character. Therefore, you cannot assume that a particular number always controls the same command.

➤ To choose an option on the Action menu using a number key, press the number associated with the menu option.

4.2. Using the Radio

VR-Engage supports voice communications over radios and intercoms. You can communicate with anyone using the radio application that is in the same exercise. There are 2 radios, each of which has 2 channels. (The radios are configurable. For details, see "4.2.4 Configuring the MAK Radio" on page 96.) Two applications can communicate only when the radios are tuned to the same frequency and that frequency is not being jammed

Ground vehicles have an intercom, which has one channel. The intercom allows the crew to communicate among themselves. For information about the intercom, see "4.2.3 Using the Intercom or a VOX Radio" on page 96.

Audio data from the radio system is sent as signal interactions/PDUs on the simulation network. It can be received in other applications that can receive standard signal interactions/PDUs.

Radio communications are perfect. Radio signals are not affected by terrain, weather, distance, or other factors. However, you can use the VR-Forces radio panel or the standalone radio application included with VR-Engage to jam radios. For more information about jamming, see "4.2.6 Jamming Radios" on page 101.

The radio uses the audio devices configured in the Audio Settings dialog box for capture and playback. See "4.5.2 Configuring Sound Effects" on page 103.

➤ **To use the radio:**

1. Open the Action menu.
2. Choose **Radio** to toggle it to one of the available radios, for example, Radio 1, Radio 2, or Intercom (if one is configured).

3. When you select a radio, the channel and frequency displays. You can toggle that setting to anther channel.
4. To talk, hold down the **T** key or, if you are controlling a human character and using a game controller, press the A button. To hear others, release the key or button.

> You cannot hear other people talking while you are pressing the talk key or button.

4.2.1 Using MAK Radio in VR-Forces

The MAK Radio feature in VR-Forces allows VR-Forces users to use the radio to talk to VR-Engage players or anyone running the radio application.

To learn more about using the MAK Radio in VR-Forces, please refer to the *VR-Forces Users Guide* or online help.

4.2.2 Using the Standalone Radio Application

VR-Engage includes a standalone radio application. It lets you communicate with VR-Engage players without running VR-Engage. The application has different versions for the protocols supported by VR-Engage.

> The default key for Push to Talk is **T**, but you can specify other keys in the Audio Settings. See "4.2.5 Audio Settings" on page 100.

➤ **To use the VR-Engage standalone radio:**

1. In a console window or File Explorer, change to *./bin64*.
2. Run the executable for the protocol you are using, as follows:

   DIS:

   ```
   makRadioAppDIS.exe
   ```

   HLA Evolved:

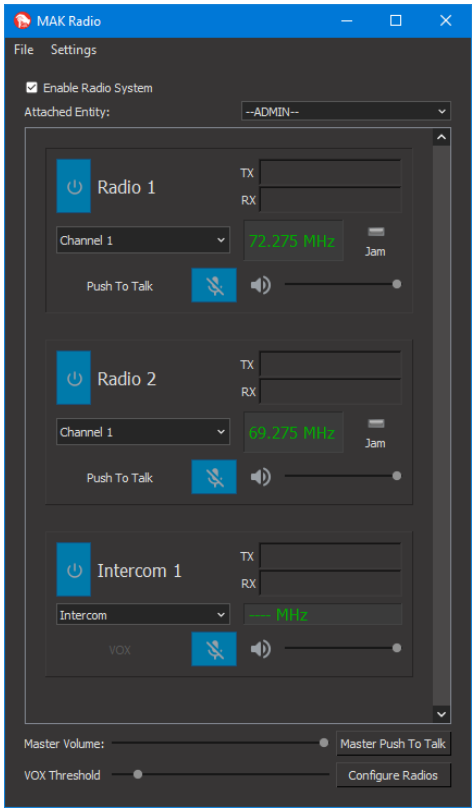
   ```
   makRadioAppHLA1516e.exe
   ```

   HLA13:

   ```
   makRadioAppHLA13.exe
   ```

   MAK Radio opens (Figure 27).

Figure 27. MAK Radio



3. Optionally, adjust the Master Volume slider to adjust the volume of all radios.
4. Optionally, adjust the VOX Threshold slider to adjust the level of sound needed to enable voice activated transmission. (This is usually used for intercoms, not radios.)
5. Optionally, click Master Push to Talk to transmit on all radios that are not muted.
6. Operate a radio head as follows:
    a. Make sure the radio head power button is enabled ( ).
    b. Optionally, select a channel from the list.
    c. Click Push to Talk or press **T** (or other key) to talk (if this is not a Vox radio).
    d. Click the microphone button ( ) to mute the microphone or enable it.
    e. Click the speaker button ( ) to mute the speaker or enable it.
    f. Adjust the slider to change the radio's volume.
    g. Click the Jam button ( ) to jam the radio frequency.
7. Click Configure Radios to edit the radio configuration. See "4.2.4 Configuring the MAK Radio" on the next page for details.

8. Optionally, choose **Settings > Audio Settings** to select the Playback Device (speaker) and Recording Device (microphone).

4.2.3 Using the Intercom or a VOX Radio

The intercom and VOX radios are voice activated and transmit any sound greater than the VOX threshold.

➤ **To use the intercom or a Vox radio:**

1. In the MAK Radio Application, select the Attached Entity.
2. In the application, make sure the power button is enabled ( ).
3. Make sure that the microphone is not muted.
4. If you are using a Vox radio, you can select a channel.
5. Start talking.

4.2.4 Configuring the MAK Radio

You can configure the number of radios and channels, the channel frequencies, the sample rate, and the format. The Modulation and Crypto settings for radios are as defined for the Transmitter PDU in the IEEE DIS standard, IEEE Std 1278.1-2012.

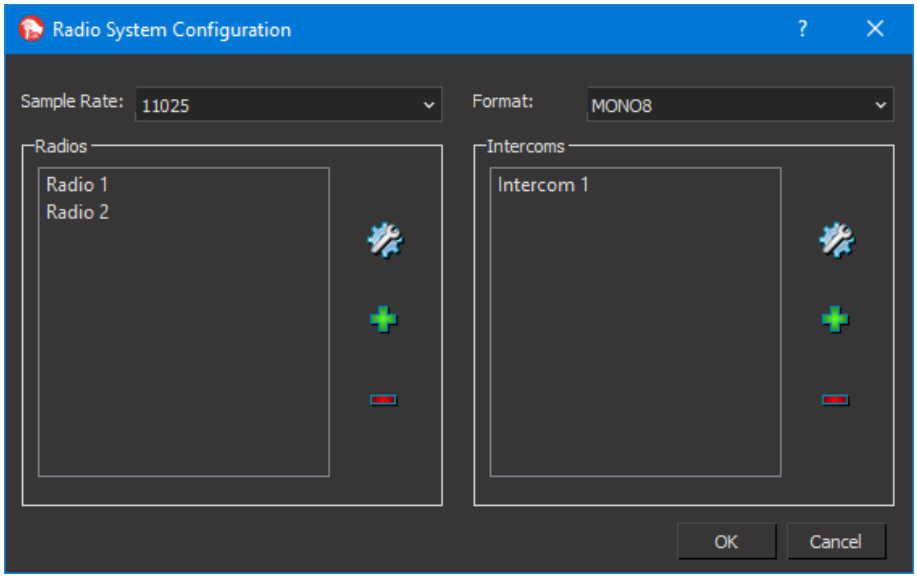
If there is a mismatch between two or more radios in the Transmission, Modulation, and Crypto settings, they cannot talk to each other. The settings must match between communicating radios for the signals to be processed and the transmissions to be heard.

Changes that you make persist from one session of VR-Engage and MAK Radio to another, and the settings are the same for both.

➤ **To configure the MAK Radios:**

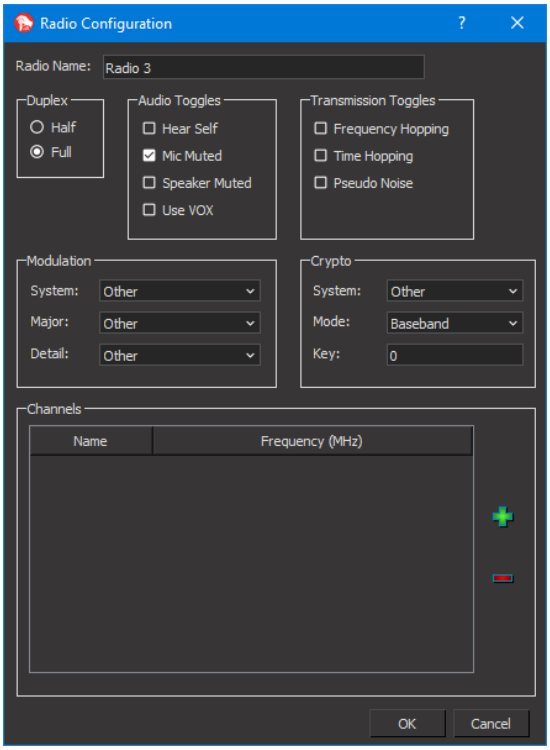
1. In the MAK Radio application, click Configure Radios. The Radio System Configuration dialog box opens (Figure 28).

Figure 28. Radio System Configuration



2. To change the sample rate for all radios, select the rate from the list.
3. To change the format for all radios, select the format from the list.
4. To add or edit a radio:
    a. Click the Add Radio () or Edit button () next to the list of radios. The Radio Configuration dialog box opens.
    b. Configure the radio as needed. Table 3 on page 99 describes the settings.

Figure 29. Radio Configuration



c. Click OK.

5. To add or edit an intercom:

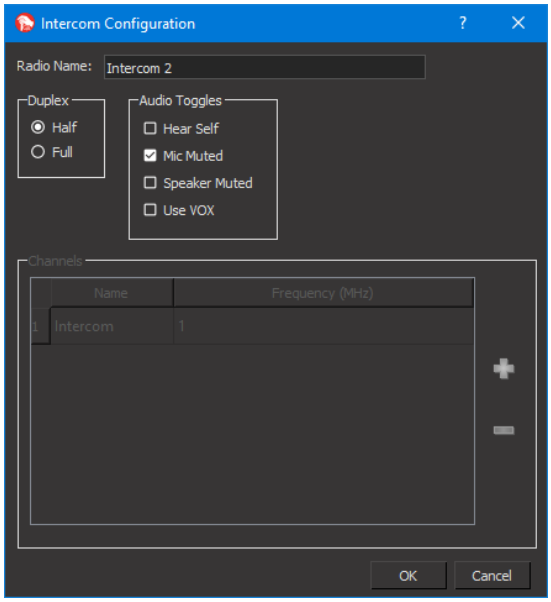
   a. To add an intercom, click the Add Intercom (). A row is added.

   b. To edit a new or existing intercom, select the row and then click the Edit button () next to the intercom list. The Intercom Configuration dialog box opens (Figure 30).

   c. If desired, edit the name of the intercom.

   d. Configure the intercom as needed. Table 3 provides information about the settings. Note that only Duplex and Audio Toggles settings are available for intercoms.

Figure 30. Intercom Configuration



e. Click OK.

6. When you are finished, click OK in the Radio System Configuration dialog box. You return to the Radio Communications dialog box. If you added or removed a radio or intercom, the list updates.

Table 3. Radio and Intercom settings

Setting	Options
Duplex	Radios and Intercoms — Half. Only one user can talk at a time. — Full. All users can speak simultaneously.
Audio Toggles	Radios and Intercoms — Hear Self. Users can hear themselves over the speaker. — Mic Muted. Microphone defaults to muted. — Speaker Muted. Speaker defaults to muted. — Use Vox. Radio is voice-activated.

Table 3. Radio and Intercom settings (continued)

Setting	Options
Transmission Toggles	Radios only — Frequency Hopping. Switches frequencies during transmission to reduce interference and avoid interception. — Time Hopping. Varies the period and duty cycle to achieve anti-jamming or low probability of intercept. — Pseudo Noise. Introduces a deterministically generated random sequence of bits.
Modulation	Radios only — System. Radio system type: Generic, SINGCGARS, and so on. — Major. Modulation type: Amplitude, Pulse, and so on. — Detail. Additional information about the modulation type (options based on the Major setting).
Crypto	Radios only — System. Cryptographic system for the radio: VINSON, KY-100, and so on. — Mode. Crypto mode: Baseband, Diphase, and so on. — Key. A simulated cryptographic key. This key is a non-zero integer value for simulating secured communication. Set to 0 when simulating plain communications.
Channels	Radios only — Use the Add or Delete buttons to manage the radio's channels. — In the rows, double-click to edit each channel's name and frequency as needed.

4.2.5 Audio Settings

In the MAK Radio standalone application, you can access the Audio Settings. These settings allow you to specify the Playback Device (speaker), Recording Device (microphone), and one or more push-to-talk buttons.

The options are saved in *./appData/settings/makRadio/radioAudioSettings.csv*. You can copy the CSV file to other workstations that should have the same settings.

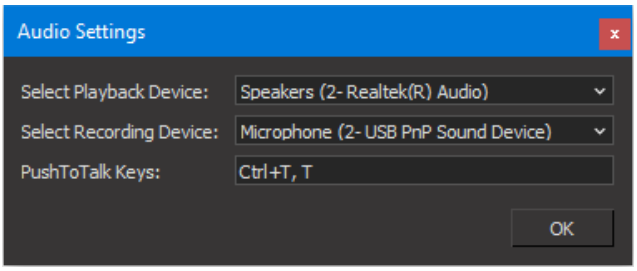
➤ **To choose a Playback Device or Recording Device:**

1. In the MAK Radio standalone application, choose **Settings > Audio Settings**. The Audio Settings dialog box opens.
2. Choose the speaker or microphone from the appropriate list.
3. Click OK.

➤ **To set the default push-to-talk key:**

1. If the Audio Settings dialog box is not open, choose **Settings > Audio Settings**.
2. Click on the Push To Talk Keys box to give it focus.
3. Press the desired key or key combination. If you want to use more than one key, continue pressing keys. Each key or key combination is separated by a comma (Figure 31).
4. Click OK.

Figure 31. Audio Settings for MAK Radio



4.2.6 Jamming Radios

The MAK Radio Application allows you to jam radio frequencies. When you click the Jam button on a radio head, it creates another radio that jams the frequency of the currently selected channel. The users will not be able to hear the radio communications, and will not see an indication that their radio is jammed.

➤ To jam a radio frequency, click the Jam button (Jam).

4.3. Changing the View Mode

By default, when you assume a role, you are in first-person view mode. That is, the eyepoint is placed as if you were looking out of the eyes of a human character or from the appropriate position of a character in a vehicle, such as the driver or pilot. Using the keyboard, you can switch to third-person view so that you can see the character you are playing in its context in the terrain. If you are driving a vehicle, you may have several views of the vehicle from different angles available.

➤ **To change the view mode, do one of the following:**

- For human characters and aircraft, press **V**.
- For ground vehicles, press **V** or a number **1** through **6**.

4.4. Starting VR-Forces Events

VR-Forces scenarios can include scenario events, which can consist of text, audio, video, graphics, or some combination of these. A scenario event represents an incident or condition that exercise participants can respond to, for example a political event, a natural disaster, or some other novel event that might require a participant to respond. Events can also be used to satisfy the conditions in a trigger.

If you are controlling a character in a VR-Forces scenario and the scenario has scenario events, you can trigger them. The scenario events are displayed in the VR-Forces front-end. They are not displayed in the VR-Engage window. If VR-Engage is hosting a scenario (and there are no VR-Forces front-ends in the exercise), the events are listed on the Action menu, but triggering them may have no visible effect.

The Action menu lists all of the events that have been created for the scenario, but does not indicate their status. A VR-Engage character can only start an event. If an event is active, you cannot stop it from VR-Engage.

➤ **To start a scenario event:**

1. Open the Action menu.
2. Choose **Trigger Event**.
3. Choose the event you want to trigger.

4.5. Introduction to Sound Effects

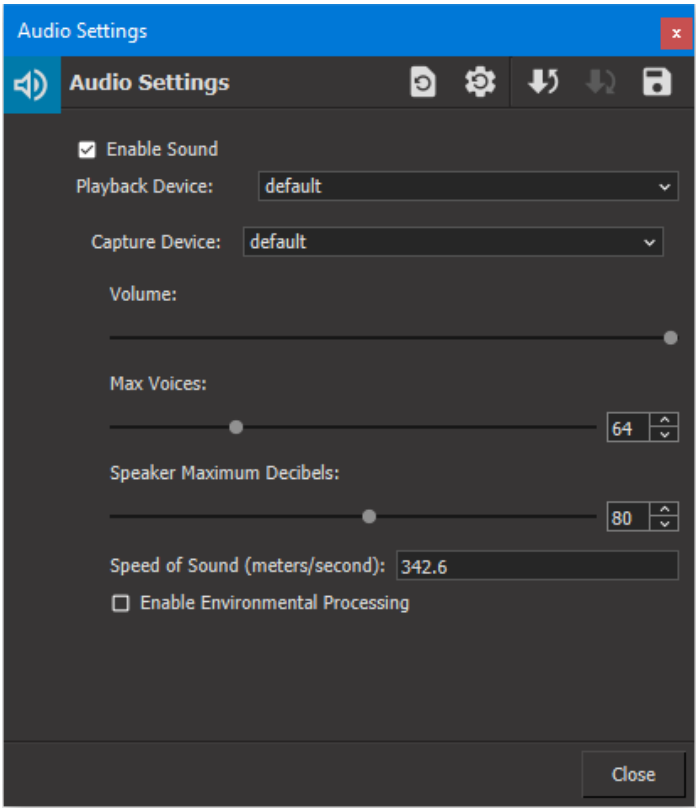
VR-Engage can associate sound effects with entities, fire events, and detonation events. You can enable and disable use of sound effects and you can specify that the player will hear sounds only when their character is within a certain distance of the entity or event that generates them.

4.5.1 Enabling Sound Effects

➤ **To enable sound effects in Admin mode:**

1. Choose **Settings > Audio Settings**. The Audio Settings dialog box opens (Figure 32).

Figure 32. Audio Settings page



2. Select the Enable Sound check box.
3. Optionally, specify the playback device (speaker) and capture device (microphone).

4.5.2 Configuring Sound Effects

You can configure the following sound settings in Admin mode:

- Playback Device. Select the speaker for the audio output.
- Capture Device. Select the microphone to use. This is typically for use with MAK Radio.
- Volume. This is the volume of the system.
- Max Voices. You can have from 1 to 256 voices, or maximum simultaneous playing sounds. Sounds that play at the same time are mixed into a single sound wave prior to being sent to the speaker. This mixing has a CPU cost. A high number of allowable voices can affect performance when there are many audio

sounds trying to play.

- Speaker Maximum Decibels. Calibrates the audio system for the speakers that the system is played on. This affects how sounds at a distance fall off so that the sound loudness is realistically replicated. The default setting of 80db represents headphones or a desktop PC's speakers. See "4.5.3 Decibels Configuration" below for more information.
- Speed of Sound. This is the speed of sound in the atmosphere in meters/second. The default value is derived by using a standard day as parameters (20 C, 50% relative humidity, 1013.25 mbar/29.92 inHg). This affects the delay in playing sounds at a distance. For example, an explosion 2km away has a several second delay from the time you can see the visual effect to when you can hear the audio effect.
- Enable Environmental Processing. Enables low pass filtering. As sound propagates through air, higher frequency sounds get absorbed more the further you move from the source of the sound. This setting lets you replicate air absorption of a sound based on the distance the sound is from the listener.

➤ **To configure sound settings:**

1. Choose **Settings > Audio Settings**. The Audio Settings dialog box opens (Figure 32 on the previous page).
2. To select the speaker, choose a device from the Playback Device list.
3. To select the microphone, choose a device from the Capture Device list.
4. To set the volume, adjust the Volume slider.
5. To specify the maximum number of voices, adjust the Max Voices slider or change the value in the text box.
6. To set the speaker maximum decibels, adjust the Speaker Maximum Decibels slider or change the value in the text box.
7. To set the speed of sound, type a value in the Speed of Sound box.
8. To enable lowpass filtering, select the Enable Environmental Processing check box.

4.5.3 Decibels Configuration

As a feature, the VR-Engage audio system uses a sound decibel scaling system to compensate for the deployed sound systems such as headphones, computer speakers, and amplifiers. This helps provide a similar audio experience regardless of the system that you use.

You can configure each individual entity sound to match real world characteristics. For example:

- A whisper or rustling of leaves registers at 20 dB.
- A conversation at home is approximately 50 dB.
- A passenger car traveling on the freeway at 50 feet away is 70 dB.
- Inside a helicopter cabin 100 dB.

When configuring the decibels (dB) for a sound, you should select a decibel level that represents the level near the sound source. The dB setting should be the loudest level that you would ever want to hear the sound, which might differ from the real-life level. This allows the audio system to calibrate the sound levels automatically.

For each sound, based on the Speaker Maximum Decibels setting, the audio system computes a sound "reference" distance and "falloff" distance:

- The reference distance is the distance at which the sound is the loudest it can be heard.
- The falloff distance is the distance at which the sound can no longer be heard.

These examples show when a VR-Engage entity's sound is configured to be 80 dB and the observer is closet to the entity:

- If the Speaker Maximum Decibels is set to 80 dB or lower, then the sound will be played and heard at its maximum volume.
- If the Speaker Maximum Decibels is set to 120 dB, then the sound will be played at a lower volume because the speaker system is expected to increase the heard volume to 80 dB.

Other sound effects settings are covered in "4.5.2 Configuring Sound Effects" on page 103.

Figure 33. Example of entity decibel setting in Entity Definition Editor

4.5.4 Associating Sound Files with Entities and Events

VR-Engage includes a set of sound files (WAV format) for generic entity platforms and for fire and detonation events. The files are in *./SharedData/16/latest/Audio/Sounds*. You add and edit mappings between entity types, fire events, or detonation events and sound files in the Visual Models Editor. Sound files are associated with model definitions. For example, the **filename** parameter for the heli_blades model definition specifies *$(SHARED_DATA_DIR)/Audio/Sounds/heli_blades.wav*. The Rotary Wing element definition has a parameter called **Entity Sound**. The value for the **model definition** attribute is **heli_blades**.

You can quickly see which model definitions are available for sounds by filtering the list of the Visual Model Editors, Model Definitions Editor page for the Sound schema.

4.6. Resetting Entities

In Admin mode, you can reset entities, which can be useful during tests or other situations. When you reset an entity, their health is restored, their resources are replenished, such as fuel and munitions, and, in the case of vehicles, they are set upright. You might want to reset the entity if you die, roll a vehicle, crash a plane, expend all of your ammunition, or are otherwise rendered unable to continue performing the desired task. To learn about Admin mode, see "3.1.3 Admin, User, and Lockdown Modes" on page 51.

There are two types of reset:

- Reset. Restores the entity and returns it to the location it had when you took control. If you are controlling a human entity that disembarked from a vehicle role, it resets to the location at which it disembarked.
- Reset in Place. Restore the entity at its current location.

> When you reset an entity to the location it had when you took control, none of the other entities in the scenario will reset. As a result, the location might no longer make sense in the context of the overall scenario.

➤ To reset an entity, press **F5** (Reset) or **Ctrl+F5** (Reset in Place).

You can also reset an entity using the Action menu.

➤ To reset an entity in Admin mode, open the Action menu and then choose **Reset** or **Reset in Place**.

4.7. HMD Controls

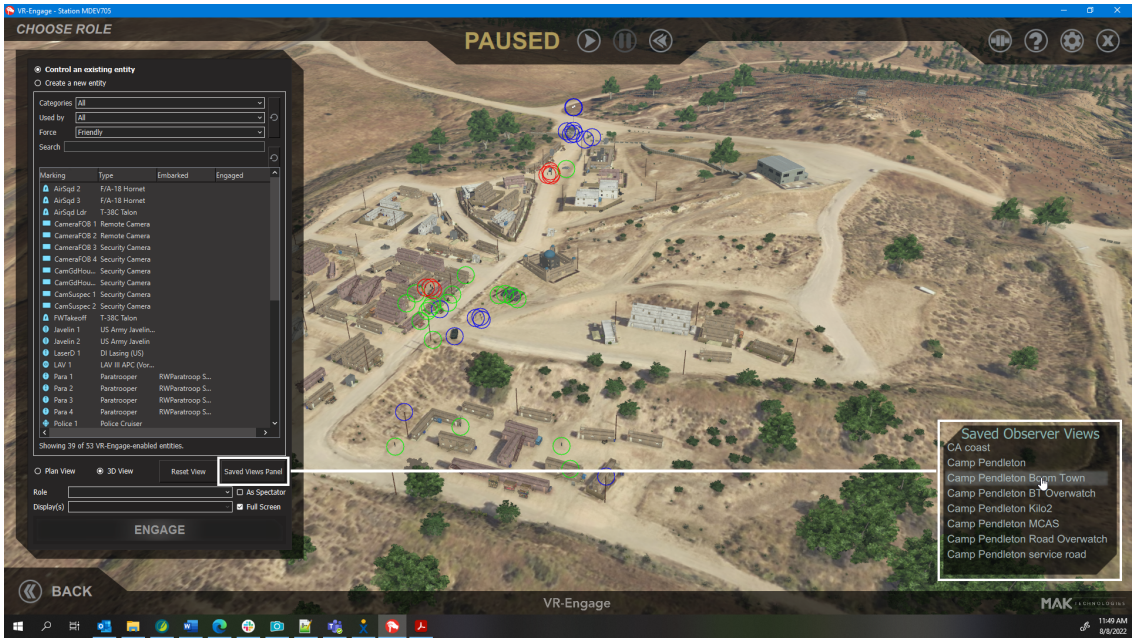
There are some minor differences for head-mounted displays.

- The Play and Pause controls are located in the Action menu.
- Pressing **Esc** returns you to the main menu and exits VR mode. You return to VR when you select Resume from the menu.

4.8. Saved Observer Views

At any given moment, the state of the observer is called its view. Most terrains included with VR-Engage include saved observer views. Saved views are listed in the Observer Views Panel, which you open by clicking Saved Views Panel in the Choose Role panel.

Figure 34. Saved Observer Views



4.8.1 Selecting a View

➤ To recall a view, select it in the Saved Observer Views panel. The view changes immediately.

4.8.2 Transitioning to a View

Normally, when you select an observer view, the observer jumps immediately to the new view. However, VR-Engage can smoothly transition from the current observer view to a saved view.

➤ To transition between observer views, hold down the **Ctrl** key and click the view in the Saved Observer Views panel that you want the observer to move to.

4.8.3 Resetting the View

You can change the observer position from the default using the standard observer keys and the mouse. You can reset to the default view for the selected observer view.

➤ To reset the view to the saved observer view, click Reset View.



5. Controlling People

This chapter describes how to control human characters and interact with the environment and other characters.

5.1. Introduction 110
5.1.1 Enabling and Disabling Mouse Control 110
5.2. Basic Movements 111
5.2.1 Moving Your Character Around the Terrain 111
5.2.2 Changing Your Character's Posture 111
5.2.3 Looking Around 111
5.2.4 Gesturing and Signaling 111
5.2.5 Climbing Ladders 112
5.3. Navigation 112
5.3.1 Using the Compass 112
5.3.2 Using the Minimap 113
5.4. Health and Stamina 114
5.4.1 Sustaining Damage to Health 115
5.4.2 Experiencing Fatigue and Losing Stamina 115
5.5. Weapons 116
5.5.1 Choosing a Weapon 116
5.5.2 Aiming a Weapon 116
5.5.3 Firing a Weapon 118
5.5.4 Firing Weapons with an HMD Controller 118
5.5.5 Throwing a Hand Grenade 118
5.5.6 Deploying and Stowing Weapons 119
5.5.7 Reloading a Weapon 120
5.6. Javelin Missile Launchers 120
5.6.1 Using the Shoulder-Mounted Missile Launcher 121
5.6.2 Using the Tripod-Mounted Missile Launcher 121
5.7. Using the AN/PED-1 LLDR 123

5.8. Embarking on a Vehicle 124
5.8.1 Embarking on the Stryker's Interior 125
5.8.2 Disembarking from a Vehicle 126
5.9. Carry or Drag a Person 126
5.10. Parachuting 127
5.11. Tasking Other Entities 127
5.11.1 Disembark Entity 128
5.11.2 Follow Me 128
5.11.3 Move to Location 128
5.11.4 Move to My Location 129
5.11.5 Suppress Location 129
5.12. Turning on Your Flashlight 130
5.13. Using the Binoculars 130
5.14. Changing Visual Sensors 131
5.15. Lasing Targets 131
5.15.1 Defining the Laser Code 132
5.15.2 Displaying a Gimbaled Sensor's View 133
5.15.3 Enabling Laser Designators 133
5.16. Opening and Closing Doors and Windows 133

5.1. Introduction

VR-Engage allow you to control individual human characters. VR-Engage includes models for several types of dismounted infantry and civilian characters, including:

- Dismounted infantry
- Dismounted infantry with lasing capability
- Dismounted infantry with missile launchers
- Paratroopers
- Police officers

5.1.1 Enabling and Disabling Mouse Control

When mouse control is enabled, you can use the mouse to turn a character, fire, and other actions. When it is disabled, you can access menus (in windowed mode) and do other typical mouse actions.

When you take control of a character, mouse control is automatically enabled. When you switch to windowed mode, it is automatically disabled. You can toggle mouse control on and off at any time in both full-screen mode and windowed mode.

➤ To enable or disable mouse control, press **Ctrl**. You can also double-click in the application to re-enable mouse control.

5.2. Basic Movements

As a player, you can move your character around and perform gestures.

5.2.1 Moving Your Character Around the Terrain

You can move your character around the terrain using the keyboard, game controller, or head-mounted display (HMD) controller. You can look around using the keyboard, game controller, HMD controller, or mouse. You can increase and decrease the speed at which you move.

5.2.2 Changing Your Character's Posture

You can change your character's posture. The default is to be standing upright, but you can also crouch or lie prone. Crouching helps you move into spaces with a low clearance or hide behind obstacles. Lying prone makes you less of a target for enemies and improves your accuracy when aiming a weapon.

An icon indicates your character's posture.

➤ **To change your character's posture:**

- Press **E** to move down to a crouching or prone position.
- Press **Q** to move up to a crouching or standing position.
- Press **Z** to lay prone from any position.
- Press **C** to stand from any position.
- Press the B button on the game controller to cycle between standing, crouching, and lying prone.

5.2.3 Looking Around

You can look around by turning only your character's head, instead of turning the entire body. While you are looking, you cannot fire a weapon.

➤ **To look without moving your body:**

1. Press and hold **.** (period) or the right thumbstick.
2. Use the mouse, the right thumbstick, or the **H**, **K**, **U**, or **J** keys to look around you.

5.2.4 Gesturing and Signaling

Your character can perform a gesture or signal, such as pointing or surrendering. Some signals have military meanings, which are only available to military characters.

➤ **To make a gesture:**

1. Open the Action menu.
2. On the Action menu, choose **Gesture**.

3. Use an Action menu control to show the submenu. The Gesture menu opens.
4. Choose a gesture.

Your character performs the gesture. People looking at your character will see the gesture. (If you are in the first-person view, you will not see yourself performing the gesture. You can see it in third-person view.)

5.2.5 Climbing Ladders

If there are ladders in the terrain, you can climb them.

➤ **To climb a ladder:**

1. Move your character to the base or top of the ladder. When your character mounts the ladder, the ladder fills the view.
2. Climb up or down the ladder:
   - To climb up the ladder, move forward.
   - To climb down the ladder, move backward.

5.3. Navigation

While playing a human character, you can use the compass and the minimap to help you navigate the terrain.

5.3.1 Using the Compass

The compass at the top allows you to see which direction your character is facing. The compass is horizontal and shows the direction where your character is looking.

While in the human role, you can choose whether the compass should show degrees or NATO-MIL. You can also choose whether you want the compass to reference True, Magnetic, or Grid north. True north, also known as geodetic north, is the direction pointing to the geographic location of the North Pole. Magnetic north, on the other hand, is the point where the magnetic field points down. In VR-Engage, it is determined by the World Magnetic Model (WMM). The current parameters are good until 2025. To learn more about the WMM, visit https://www.ngdc.noaa.gov/geomag/WMM. Lastly, grid north is the direction indicated by the vertical grid lines on ordnance survey maps.

Figure 35. Compass



➤ **To enable or disable the compass:**

1. Open the Action menu (Figure 26 on page 92).
2. On the Action menu, choose **Compass**.
3. Use an Action menu control method to toggle the compass on or off.

➤ To toggle the compass on or off, you can also press **G**.

➤ **To change the units displayed in the compass:**

1. Open the Action menu.
2. Use an Action menu control method to toggle the Compass Units to either NATO-MIL or Degrees.

➤ **To specify which north the compass should use:**

1. Open the Action menu.
2. Use an Action menu control method to toggle the North Reference to True, Magnetic, or Grid.

The default settings for the compass are based on the 3D Compass and Heading specified in the Display Units Settings; see "Specifying Display Units" in the *VR-Vantage Users Guide* to learn how to adjust these settings while in Admin mode.

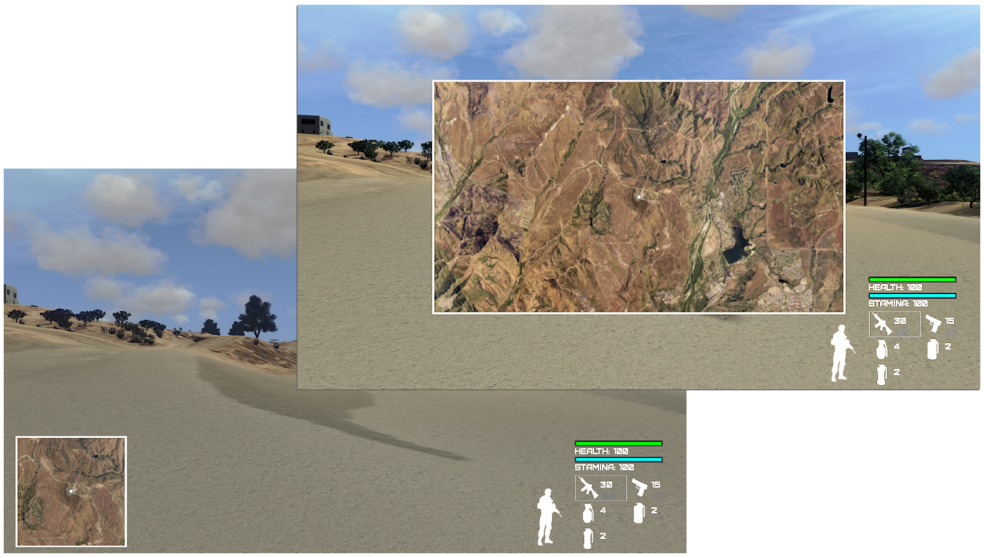
5.3.2 Using the Minimap

You can open a minimap that shows where your character is in relation to the terrain.

The minimap is hidden by default. Use the Action menu or the keyboard to select the display format. Using the keyboard, you can also zoom in or out of the minimap.

You can configure the available map sizes and the minimap's border in *./data/simulationModelSets/VR-Engage/role/devices/minimap.lua*.

Figure 36. Minimap - corner and center



➤ **To change the minimap display using the Action menu:**

1. Open the Action menu (Figure 26 on page 92).
2. On the Action menu, choose **Minimap**.
3. Use an Action menu control method to toggle the setting: hidden, corner, or center.

➤ **To change the minimap display using the keyboard:**

- Press **M** to cycle through the corner, center, and hidden views.
- Press **+** on the number pad to zoom in on the minimap.
- Press **-** on the number pad to zoom out of the minimap.
- Press **0** on the number pad to reset the minimap to the default zoom.

5.4. Health and Stamina

The human entity overlay includes bars that indicate the current health and stamina for your character (Figure 37). Health and stamina are depleted as your character sustains damage and experiences fatigue. Tracking human entities' health and stamina adds realism to your simulation.

Figure 37. Health and stamina bars



5.4.1 Sustaining Damage to Health

Unless your character is set to be invulnerable, you will sustain damage when other characters attack you. The health bar indicates the state of your health (Figure 37 above). As you take damage and your health is depleted, you will have trouble moving around and aiming weapons. After a certain amount of damage, you fall to the ground in pain. At that point, your view lowers to ground level, and you cannot move or aim a weapon. Other characters in the scenario also sustain damage and experience the same effects (if they are not invulnerable).

The location on the body where a character is injured determines how much damage a character sustains. For example, a shot in the leg or arm is less lethal than a shot to the torso or head.

As you take damage, the screen shows some red. When you are hit from the side, the corners of that side of the screen are highlighted in red. If you are shot from the front or the back, all corners of the screen are highlighted. In addition, as your health is depleted, the screen takes on a red tint and your aiming will be less accurate.

5.4.2 Experiencing Fatigue and Losing Stamina

Sprinting, crawling, walking in a crouched position, and carrying or dragging a wounded human are more challenging and therefore they cause you to use your stamina and experience fatigue. The stamina bar displays your current level of stamina (Figure 37 above). As you grow fatigued and your stamina reserves are reduced, you cannot move quickly and you will also see a greater margin of error when you fire a weapon.

When you walk or stop moving, you regain your stamina. You recover faster if you stop moving than if you walk.

5.5. Weapons

Infantry characters have a rifle, a handgun, and several types of grenades. The name indicates a character's primary weapon; for example, characters with an M4 in their name have an M4 rifle. A graphic in the lower right corner of the screen shows the current weapon (Figure 38).

Figure 38. Weapon overlay for humans



5.5.1 Choosing a Weapon

➤ To cycle through the weapon list, use the Select Weapon button on your game controller or press the **Spacebar**.

➤ To choose a specific weapon using the keyboard, press the number key that correlates to the weapon.

➤ **To choose a weapon using the Action menu:**

1. Open the Action menu (Figure 26 on page 92).
2. On the Action menu, choose **Weapon**.
3. Use an Action menu control method to show the submenu. The Weapons menu opens.
4. Choose a weapon.

5.5.2 Aiming a Weapon

You can aim a weapon using the reticle or the sight (Figure 39 and Figure 40).

The reticle indicates the accuracy of your shots. If you are moving, fatigued, or wounded, there is a greater margin of error in your accuracy and the reticle increases in size. To improve your accuracy, you can hold still or change to a prone position. The reticle shrinks to indicate a lesser margin of error.

Figure 39. Aiming with the reticle



Figure 40. Aiming with sight



➤ To aim a weapon using the reticle, turn and raise or lower the weapon using the mouse (in mouse-look mode) or the right thumbstick on the game controller, or by pressing the **H**, **K**, **U**, or **J** keys.

➤ **To aim a weapon using the sight:**

1. Hold down the right mouse button (in mouse-look mode), the left trigger on your game controller, or the **,** (comma) key.
2. Aim the sight using the mouse or the right thumbstick, or by pressing **H**, **K**, **U**, or **J**.

5.5.3 Firing a Weapon

Most guns shoot once each time you fire the weapon. The AK-47 fires in three-shot bursts.

➤ **To fire a weapon, do one of the following:**

- Click the left mouse button (in mouse-look mode).
- Press the right trigger on your game controller.

5.5.4 Firing Weapons with an HMD Controller

If you are using a head-mounted display (HMD) with hand controllers, you can fire weapons using your controller. By default, the right hand controller is the dominant controller. You can change this by changing the **rightHanded** parameter in *./data/simulationModelSets/VR-Engage/roles/displayLayouts/humanVrLayouts.lua* from true to false.

- If you have a pistol, hold the controller in front of you to aim. Press the trigger to fire. The pistol only tracks the position of the dominant hand.
- If you have a rifle, align the two controllers as if you were holding a rifle, with the dominant hand closer to your body. Aim the rifle and fire.

Figure 41. Firing a rifle using HMD hand controllers



5.5.5 Throwing a Hand Grenade

Soldiers can throw fragmentation grenades, smoke grenades, and flash bang grenades. Grenades are thrown at a target point. (If the soldier is aiming towards an area with a light background, the target point may be hard to see.)

Figure 42. Grenade target point



➤ **To throw a grenade:**

1. Use the mouse or the right thumbstick to aim the grenade.
2. Click the left mouse button (in mouse-look mode) or press the right trigger on your game controller.

To use the keyboard to aim the grenade, press the **H**, **K**, **U**, or **J** keys.

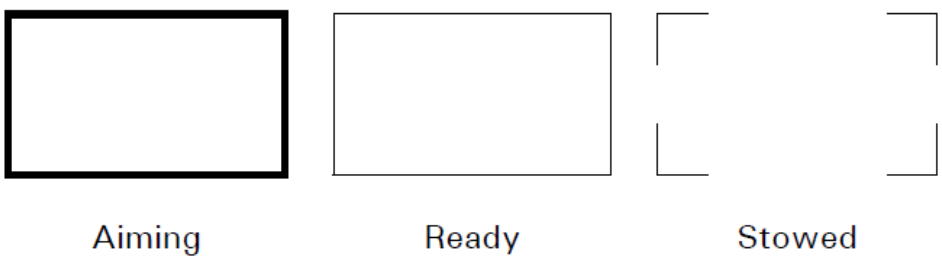
5.5.6 Deploying and Stowing Weapons

When a weapon is deployed (Aiming state), you see it in first-person view and you can fire it. When it is in the Stowed or Ready state, you cannot see it in first-person view and you cannot fire it.

The weapon overlay indicates the status of the weapon, as follows and as illustrated in Figure 43:

- Aiming. Weapon icon has a thick rectangle around it.
- Ready. Weapon has a thin rectangle.
- Stowed. Weapon has a bracketed rectangle.

Figure 43. Weapon status



➤ **To deploy or stow a weapon:**

1. Open the Action menu.
2. On the Action menu, choose **Weapon State**.
3. Use an Action menu control method to toggle between states.

➤ To cycle the weapon state, press ' (apostrophe).

➤ To move a weapon from the Ready state to the Aiming state, press the left mouse button or right trigger on the controller.

5.5.7 Reloading a Weapon

As you fire a weapon, the system tracks the amount of ammunition left in the magazine. When the magazine is depleted, your character automatically pauses to reload. At any point, you can also manually reload the weapon. The character reloads a magazine's worth of ammunition.

Figure 44. Magazine and ammunition counts



➤ **To reload your weapon with a fresh magazine, do one of the following:**

- Press **R**.
- Open the Action menu and then choose **Reload**.

5.6. Javelin Missile Launchers

VR-Engage has two characters with Javelin missile launchers, one with a shoulder-mounted Javelin missile launcher (US Army Javelin) and one that can deploy a tripod-mounted Javelin (US Army Javelin (Tripod)).

5.6.1 Using the Shoulder-Mounted Missile Launcher

➤ **To use the shoulder-mounted missile launcher:**

1. Add or select a US Army Javelin character.
2. Engage the character.
3. On the Action menu, set the Missile Launcher option to Aiming. The display changes to show the soldier in third person view, and also to add an aiming reticle and a view finder display to the screen.

Figure 45. Shoulder-mounted Javelin



4. Optionally, press **B** or the X button on your game controller to expand the viewfinder to cover the entire screen. (You can also hide it entirely.)
5. Fire the weapon using the standard weapon firing methods.

> The character can walk around with the weapon in the Aiming state (third-person view). You can also change the Missile Launcher state to Deployed and walk around in first-person view. In first-person view, the missile launcher is not visible.

5.6.2 Using the Tripod-Mounted Missile Launcher

The tripod-mounted missile launcher is an object that the US Army Javelin (Tripod) character can deploy. Until it is deployed, it is not part of the scenario. Once it is deployed, it is an independent entity in the scenario. Other characters can fire it. Only the character that originally deployed it can recover it; that is, remove it from the scenario.

A tripod-mounted missile launcher has an azimuth (horizontal range) of 45 degrees to the left and right. It has an elevation (vertical range) of 30 degrees. When a tripod-mounted missile launcher is deployed, it faces in the direction of the entity that deployed it. Once it is deployed, you cannot change its heading.

➤ **To deploy the tripod-mounted Javelin:**

1. Add or select the US Army Javelin (Tripod) character.
2. Engage the character. The default presentation is that of a US Army M4 character.
3. Walk to the location where you want to deploy the missile launcher.
4. Face the heading that you want the missile launcher to be deployed to.
5. Optionally, stow your rifle.
6. On the Action menu, choose **Deployable Objects > Deploy Javelin Launcher**. The Javelin Missile Launcher is added to the scenario. The Action menu changes to show Javelin Launcher menu options.

Figure 46. Tripod-mounted Javelin



Firing a Tripod-Mounted Missile Launcher

Once a missile launcher is deployed, any character can fire it.

➤ **To fire the missile launcher:**

1. If your character is close to the missile launcher, on the Action menu, choose **Javelin Launcher > Use Launcher**.

   If your character is not close the missile launcher, walk towards it until the Javelin Launcher menu displays and then choose **Use Launcher**. The display changes to show the view finder and reticle for the launcher. You are now in direct control of the missile launcher instead of the character you were controlling.

Figure 47. Javelin tripod - aiming



2. Optionally, press **B** or the X button on your game controller to expand the viewfinder to cover the entire screen.
3. Aim the Javelin and fire it.
4. To give up control of the Javelin, on the Action menu, choose **Exit Launcher**. You are now in control of your character again and the Javelin is still deployed. Your weapon is stowed.

Recovering a Tripod-Mounted Missile Launcher

When you recover a tripod-mounted Javelin, you remove it from the scenario. It is not available for use until it is deployed again. Only the character that deployed the Javelin can recover it.

➤ To recover the missile launcher, choose **Javelin Launcher > Recover Launcher** on the Action menu. If you are not within a few feet of the missile launcher, the Javelin Launcher menu option is not available. If you walk towards the missile launcher it opens automatically.

5.7. Using the AN/PED-1 LLDR

You can use the AN/PED-1 LLDR (Lightweight Laser Designator Rangefinder AN/PED-1) in VR-Engage for forward observer training. A player operating the LLDR can look around, lase targets, use the radio, change the camera on the LLDR,

The device is a standalone entity, not attached to any other character. You can include it in a scenario that includes human characters for them to use, or create it as an entity and operate it directly from the Choose Role panel.

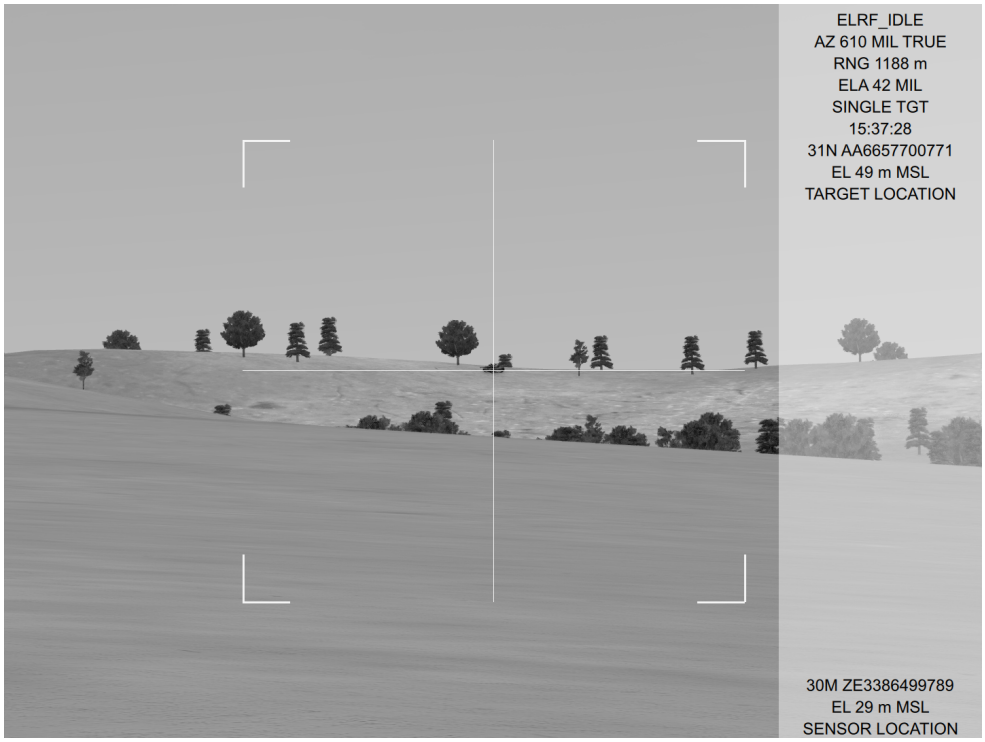


You can use the LLDR with VR devices.

➤ **To operate the LLDR as a human character:**

1. Approach the AN/PED-1 LLDR. When the LLDR menu opens, use an Action menu command to choose **Use LLDR**. Your view changes to the LLDR HUD (Figure 48).
2. Move the view to a point of interest.

Figure 48. LLDR HUD



➤ To lase a target, Open the Action menu and toggle Lasing to On. The HUD indicates that you are lasing a target: the top line changes from ELRF_IDLE to GET TGT LOCATION.

➤ **To define a laser code:**

1. In the Action menu, choose **Laser Code**. The Set Laser Code dialog box opens.
2. Enter a laser code and then press **Enter**.

5.8. Embarking on a Vehicle

If a vehicle supports embarkation, characters can embark on it and move with it as it moves. If the vehicle has interior geometry, your character can look around and walk around on the floor of the vehicle. Among the vehicles provided with VR-Engage, the 5-ton truck, the Stryker, the police car, and the CH-47 Chinook and CH-149 Cormorant helicopters have interior geometry. Other vehicles, such as the LAV and M1A2, support embarkation but do not have interior geometry.

➤ **To embark on a vehicle:**

1. Have your character walk up to the vehicle. When the character is within a meter or so of the vehicle, the embarkation menu for the vehicle displays.

Figure 49. Embark menu



2. Select a role or seat.
3. Press the **Right arrow** key or right D-pad button to initiate the action. The character is placed on the vehicle.

5.8.1 Embarking on the Stryker's Interior

The M1126 Stryker ICV with M2 has an interior that allows you to board and choose an available seat.

➤ **To embark on the Stryker from the interior:**

1. Have your character walk up to the back of the vehicle. When the character is within a meter or so of the vehicle, the Stryker menu displays.
2. If the door is not open, select **Door**. The door opens, providing a ramp into the Stryker.
3. Walk your character up the ramp and into the interior. Your character automatically shifts to a crouching posture.

   As you navigate the interior, the seat closest to your character displays in the Stryker menu. Different seats display in the menu based on your position.
4. Choose the seat from the menu. This can be a seat in the back, the Driver Seat, or the Commander seat. (You can also embark as the driver or commander directly from the Stryker menu as described in "5.8. Embarking on a Vehicle" on the previous page.)

5.8.2 Disembarking from a Vehicle

There are a couple of ways that you can disembark a vehicle.

➤ **To disembark from a vehicle, do one of the following:**

- On the Action menu, choose Disembark. The character is placed off the back of the vehicle.
- Have the character walk off the back of the vehicle.

5.9. Carry or Drag a Person

In some situations, it is necessary to carry or drag a person, such as someone who is injured. VR-Engage allows you to replicate that in a scenario. If there is a vehicle in a scenario, you can embark the person you are carrying on that vehicle.

Figure 50. Drag and carry



- If a human character is dead, you cannot carry or drag them.
- The ability to drag or carry an entity is currently limited to some of the US soldier entities.

➤ **To drag or carry a character:**

1. Approach the injured person.
2. Open the Action menu.
3. Select the character and then press the **Right arrow** key or right D-pad button.
4. Choose the method you want to use:
   - Armpit Drag. Put your hands under the injured person's armpit to move them.
   - Strap Drag. Take the injured person by the strap on the back of their vest to move them.
   - Fireman Carry. Put the injured person over your shoulder to carry them.

   The camera switches to third-person view.

5. Move in the direction where you want to take the injured person.
6. At any point, you can open the Action menu and choose **Drop Injured** to place the person on the ground.

➤ To place the person in a vehicle, choose **Embark Injured** from the Action menu.

5.10. Parachuting

The paratrooper character can parachute out of aircraft. The descent starts in free fall. The paratrooper's weapon is hidden. After the parachute is deployed, the drift of the paratrooper is affected by wind speed and direction. You can turn the parachute to the left or right and you can slow the rate of descent. When the paratrooper lands, the parachute is hidden and the paratrooper's weapon is again available. It returns to whichever view mode it was in before it jumped.

Figure 51. Parachute deployed



➤ To jump from an aircraft, walk out of an open cargo door. If the door is not open, you can open it.

➤ To deploy a parachute, press **Spacebar** or the A button on your controller.

➤ To turn left or right while the parachute is deployed, press **A** or **D**, or the shoulder buttons on the game controller.

➤ To slow the rate of descent while the parachute is deployed, press **A+D** or both shoulder buttons simultaneously.

5.11. Tasking Other Entities

While you are playing a character, you can assign tasks to other entities in the scenario.

When you assign a task, you are assigning what VR-Forces calls an independent task. If the entity to which you are assigning the task is currently executing a task either as an independent task or as part of a plan, when you assign the new independent task, the entity abandons the old task or plan.

5.11.1 Disembark Entity

The Disembark Entity task causes the selected entity to disembark from the vehicle that it is on.

➤ **To issue the Disembark Entity task:**

1. Open the Action menu.
2. Choose **Task Entity**.
3. Use an Action menu control method to show the submenu. The Select Entity menu opens.
4. Select the entity that you want to give a task to.
5. Use an Action menu control method to show the submenu. The Task Entity menu opens.
6. Choose **Disembark Entity**.
7. Use an Action menu control method to initiate the action.

5.11.2 Follow Me

The Follow Me task causes the selected entity to follow your character.

If you task an entity to follow you, it will do so indefinitely. To get it to stop following, task the entity to move to a location.

➤ **To issue the Follow Me task:**

1. Open the Action menu.
2. Choose **Task Entity**.
3. Use an Action menu control method to show the submenu. The Select Entity menu opens.
4. Select the entity that you want to give a task to.
5. Use an Action menu control method to show the submenu. The Task Entity menu opens.
6. Choose **Follow Me**.
7. Use an Action menu control method to initiate the action. The entity moves to a location behind your character.

5.11.3 Move to Location

The Move to Location task causes the selected entity to move to the specified location.

➤ **To issue the Move to Location task:**

1. Open the Action menu.
2. Choose **Task Entity**.
3. Use an Action menu control method to show the submenu. The Select Entity menu opens.
4. Select the entity that you want to give a task to.
5. Use an Action menu control method to show the submenu. The Task Entity menu opens.
6. Choose **Move to Location**. VR-Engage places a waypoint symbol at the point your character is looking at.
7. Press and hold the right mouse button or use the right thumbstick on your game controller to move the waypoint to the location you want the other entity to move to.
8. Use an Action menu control method to initiate the task. The tasked entity moves to the location.

5.11.4 Move to My Location

The Move to My Location task causes the selected entity to move so that it is standing next to your character.

➤ **To issue the Move to My Location task:**

1. Open the Action menu.
2. Choose **Task Entity**.
3. Use an Action menu control method to show the submenu. The Select Entity menu opens.
4. Select the entity that you want to give a task to.
5. Use an Action menu control method to show the submenu. The Task Entity menu opens.
6. Choose **Move to My Location**.
7. Use an Action menu control method to initiate the action. The selected entity moves to a position next to your character.

5.11.5 Suppress Location

The Suppress Location task causes the selected entity to fire on the specified location to suppress any entities at that location.

➤ **To issue the Suppress Location task:**

1. Open the Action menu.
2. Choose **Task Entity**.

3. Use an Action menu control method to show the submenu. The Select Entity menu opens.
4. Select the entity that you want to give a task to.
5. Use an Action menu control method to show the submenu. The Task Entity menu opens.
6. Choose **Suppress Location**. VR-Engage places a waypoint symbol at the point your character is looking at.
7. Press and hold the right mouse button or use the right thumbstick on your game controller to move the waypoint to the location you want the other entity to suppress.
8. Use an Action menu control method to initiate the task. The tasked entity fires at the location.

5.12. Turning on Your Flashlight

The flashlight casts a circle of light on the terrain.

➤ **To turn on the flashlight:**

1. Open the Action menu.
2. Choose **Flashlight**.
3. Use an Action menu control method to toggle the flashlight on or off.



You can also toggle the flashlight on and off by pressing **F**.

5.13. Using the Binoculars

You can zoom in and out in five fixed steps. At the highest zoom level (binoculars), VR-Engage displays a range finder graphic as an M22 binoculars reticle with 10 mil markers.

➤ **To zoom the view in and out, do any of the following:**

- Scroll the mouse wheel forward and back.
- On the Action menu, choose **Binoculars** and change the status. Binoculars go immediately to the highest zoom level.
- Press **B** (for binoculars view). Binoculars go immediately to the highest zoom level. Press - to zoom out or = to zoom in. The view zooms one step for each key press
- Press the X button. Binoculars go immediately to the highest zoom level. Use the left shoulder button to zoom out or the right shoulder button to zoom in.
- Press =, right shoulder button, or scroll the mouse wheel forward. Binoculars go immediately to the highest zoom level.

You cannot fire a weapon when the binoculars are in use. If you are controlling a character that has lasing capability, pressing the Fire button lases the target.

5.14. Changing Visual Sensors

Human characters can simulate the effect of wearing night vision goggles (NVG).

> The default visual sensors use CameraFX, which just applies a filter to the scene. It does not take into account the materials of the viewed objects. For details, see the *VR-Vantage Users Guide*.
>
> If you want a visual sensor capability that takes into account the materials of the viewed objects, you can install SensorFX. SensorFX is an extra cost option. Please contact info@mak.com for details.

➤ **To turn on the night vision goggles for human characters:**

1. Open the Action menu.
2. Choose **Night Vision**.
3. Use an Action menu control method to toggle night vision on or off.

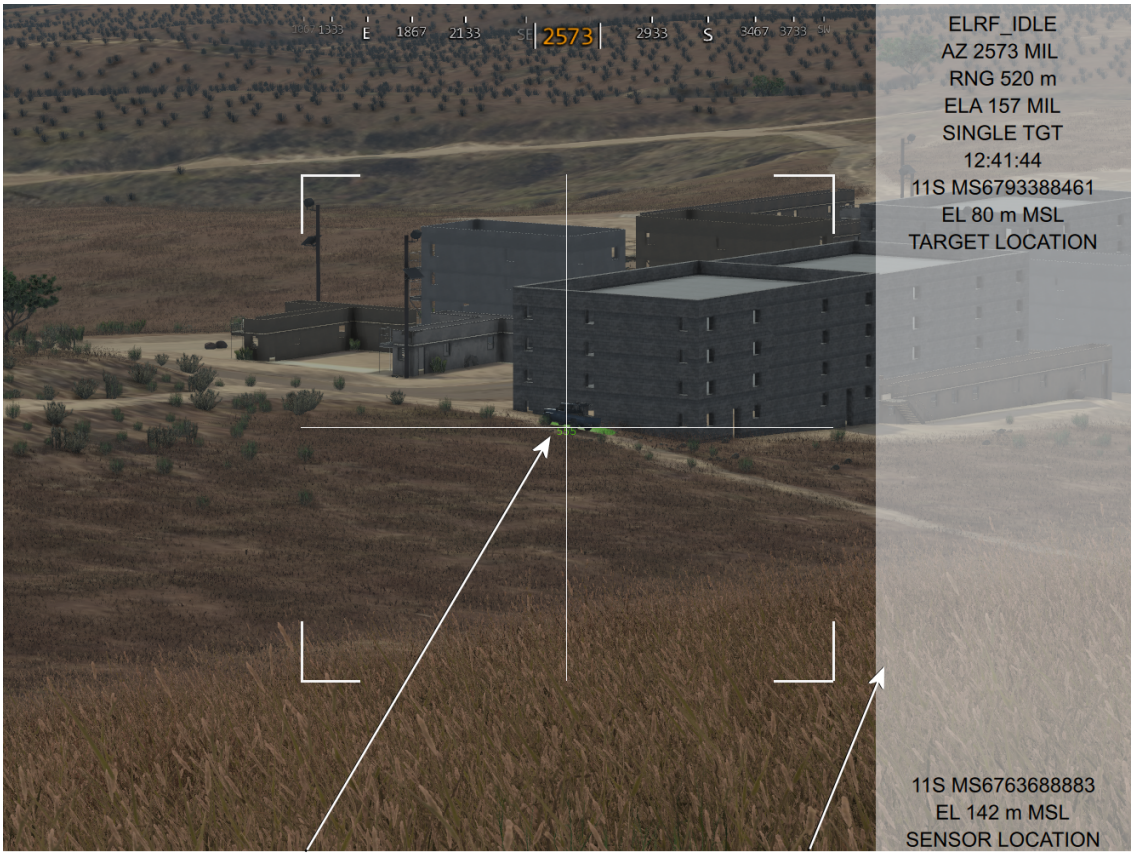
> You can also toggle night vision by pressing **N** or D-pad down on a game controller. You do not need to open the Action menu.

5.15. Lasing Targets

Soldiers that have a lasing capability can lase targets. VR-Engage includes two characters with lasing capability: DI Lasing (US) and DI Lasing (CIS).

When you lase a target, the display shows that the laser is on. If you enable laser designators, a green circle displays on the lased location.

Figure 52. Lasing a target



You cannot lase entities. You can only lase locations on the terrain.

➤ **To lase a target:**

1. Do one of the following to enable binoculars:
   - On the Action menu, toggle **Binoculars** to On.
   - Press the X button on a game controller.
   - Press **B**.
2. Point the cross-hairs of the binoculars at the point you want to lase.
3. Press and hold the left mouse button or the right trigger on a game controller.

5.15.1 Defining the Laser Code

You can define the laser code used when you lase a target. Laser codes can be up to four digits long. When you lase a target, that code is used.

➤ **To define the laser code:**

1. On the Action menu, choose **Forward Air Control**.
2. In the Forward Air Control menu, choose **Laser Code**. The Set Laser Code dialog box opens.
3. Enter a laser code and then press **Enter**.

5.15.2 Displaying a Gimbaled Sensor's View

You can display the sensor view for gimbaled sensors in the scenario. You cannot control the view, only see it.

➤ **To view the sensor on another entity:**

1. On the Action menu, choose **Forward Air Control**.
2. On the Forward Air Control menu, choose the entity whose sensor view you want to display.

5.15.3 Enabling Laser Designators

A laser designator is a green circle that shows where a laser is being pointed. It includes the laser code. Laser designators are disabled by default. You must be in Admin mode to enable laser designators.

➤ **To enable or disable laser designators:**

1. If you are not in Admin mode, press **Shift+F12**.
2. Choose **Settings > Laser Designators**.

5.16. Opening and Closing Doors and Windows

If a terrain has doors and windows that can open and close, when a human character approaches within about a meter of an opening, a menu is displayed that lets you choose to open or close the opening.

Figure 53. Door menu



The ability to open and close doors requires a dynamic terrain server. Therefore, it only works when you are connected to a VR-Forces session.

➤ To open or close a door or window, when the appropriate menu displays, make a selection from the menu.



6. Controlling Ground Vehicles

This chapter explains how to control and crew ground vehicles.

6.1. Introduction 136
6.1.1 Ground Vehicle Roles 136
6.1.2 Ground Vehicle Display Options 137
6.2. Driving Ground Vehicles 137
6.2.1 Starting the Engine 137
6.2.2 Putting the Vehicle into Gear 137
6.2.3 Controlling the Throttle 138
6.2.4 Steering a Vehicle 138
6.2.5 Releasing and Engaging the Parking Brake 138
6.2.6 Locking and Unlocking the Differential (LAV Only) 138
6.2.7 Turning the Headlights On and Off 139
6.3. The Gunner Role 139
6.3.1 Firing a Vehicle's Gun 139
6.3.2 Choosing a Weapon 139
6.3.3 Moving the Turret 140
6.3.4 Stabilizing a Vehicle's Gun 140
6.4. The Commander Roles 140
6.4.1 Overriding the Gunner 141
6.4.2 Slewing the Gun to Designate a Target 141
6.4.3 Using the Laser Designator 141
6.5. Using the Intercom 141
6.6. Changing the Sensor for a Vehicle Crew Member 142
6.7. Pivoting a Tank 142
6.8. Emergency Lights and Sirens 142
6.9. Configuring Ground Vehicles on Geocentric Terrains 142

6.1. Introduction

VR-Engage allows you to control ground vehicles that use the Vortex physics libraries, which provide high-fidelity movement behaviors.

For a list of vehicles included with VR-Engage, see "Chapter 12. Controllable Entities" on page 239. You can add more ground vehicles. For details, see "Chapter 13. Adding and Editing VR-Engage Entity Models" on page 275.

6.1.1 Ground Vehicle Roles

All ground vehicles support the Driver role, and some vehicles support other roles that you can select from the Choose Role Panel.

Some vehicles also allow you to embark as a passenger in a particular seat when you walk up to them as a human character, but not from the Choose Role Panel. See "5.8. Embarking on a Vehicle" on page 124 for details.

To learn what roles are available for each vehicle, see "Chapter 12. Controllable Entities" on page 239.

The driver role allows you to:

- Drive the vehicle, which involves:
  - Starting the engine.
  - Engaging gears.
  - Increasing and decreasing throttle.
  - Steering.
- Manage headlights.

In addition to the driver functions, the driver/gunner role allows you to:

- Rotate the turret. (LAV and M1A2 only. You cannot see this happen in the first person view.)
- Fire weapons that are configured on the vehicle. (LAV and M1A2 only. For details, see "6.3.1 Firing a Vehicle's Gun" on page 139.)

The gunner role allows you to:

- Fire weapons that are configured on the vehicle.
- Rotate a turret and raise and lower the gun. (The M2A3 Bradley IFV and GTK Boxer allow you do perform these actions while VR-Forces drives the vehicle.)
- Stabilize the gun.

The commander role allows you to:

- Look around.
- Change the sensor view.
- Zoom in and out.

- Slew or take full control of the gun (RWS only).
- Designate a target with a laser (RWS only).

6.1.2 Ground Vehicle Display Options

Most ground vehicles are designed for a single screen display. Some vehicles have support for a 3D interior or VR headsets.

- 1 Screen Horizontal. This is the standard interface for most ground vehicles.
- 3D Cockpit. This display shows a more detailed view of the interior for some roles.
- VR Headset. Select this option if you are using a head-mounted display and have chosen to engage one of the vehicles that supports VR. (This is relevant only if you have an HMD connected.)

➤ **To choose a display option for vehicles with multiple displays:**

1. In the Choose Role Panel, select the vehicle you want to control.
2. Choose the Role.
3. In the Display(s) list, select the display option. Different displays are available only on some vehicles and only for particular roles.

6.2. Driving Ground Vehicles

To operate most ground vehicles, you must use a game controller or racing wheel controller. (You can control the LAV from the keyboard in the driver role.)

6.2.1 Starting the Engine

➤ **To start (or stop) the engine, do one of the following:**

- On the Action menu, toggle Engine: On/Off.
- On a game controller, press the Start button.
- On a Racing Wheel Controller, press the Engine On button.
- Press **X**.

6.2.2 Putting the Vehicle into Gear

Vehicles have automatic transmissions. When you set a gear, you are setting the top gear it can use. Vehicles have different sets of gears. For example, the LAV has eight forward gears and one reverse gear. Automobiles have six forward gears and one reverse gear. The M1A2 tank has two forward gears and one reverse gear.

➤ **To put the vehicle into gear, do one of the following:**

- Press the right shoulder button (up) or left shoulder button (down) buttons on your game controller.
- Press **Q** (up) or **E** (down).

6.2.3 Controlling the Throttle

If you are driving using a Racing Wheel Controller, use the gas and brake pedals.

➤ **To move the vehicle, do one of the following:**

- On a game controller, press the throttle control (left thumbstick) forward.
- On a Racing Wheel Controller, depress the Throttle.
- Press **W**.

➤ **To brake the vehicle, do one of the following:**

- On a game controller, pull the throttle control backwards.
- On a Racing Wheel Controller, depress the Brake.
- Press **S**.

6.2.4 Steering a Vehicle

➤ **To steer the vehicle, do one of the following:**

- Use the right thumbstick on a game controller
- Use the steering wheel of a Racing Wheel Controller.
- Press **A** (left) or **D** (right).

6.2.5 Releasing and Engaging the Parking Brake

➤ **To set the parking brake on or off:**

- On the Action menu, toggle Parking Brake: On/Off.
- On a game controller, use left thumbstick (press).
- On a Racing Wheel Controller, press the Parking Brake button.
- Press **P**.

6.2.6 Locking and Unlocking the Differential (LAV Only)

A differential allows the wheels on an axle to rotate at different speeds. Locking the differential may improve traction under certain terrain conditions.

➤ **To lock or unlock the differential, do one of the following:**

- On the Action menu, toggle Differential Lock: On/Off.
- On a game controller, press the X button.
- On a Racing Wheel Controller, press the Differential Gear button.
- Press **C**.

6.2.7 Turning the Headlights On and Off

➤ **To turn on headlights, do one of the following:**

- On the Action menu, toggle Headlights: On/Off.
- On a game controller, press the A button.
- On a Racing Wheel Controller, press the Lights button.
- Press **L**.

6.3. The Gunner Role

The gunner can:

- Fire the gun.
- Select a munition to fire.
- Rotate and elevate the gun.
- Zoom in and out.
- Change the sensor. (For details, see "6.6. Changing the Sensor for a Vehicle Crew Member" on page 142.)

6.3.1 Firing a Vehicle's Gun

As a gunner, you can fire the weapon. Some vehicle's weapons have different munitions and you can cycle through those to select the munition you want to fire, as described in "6.3.2 Choosing a Weapon" below.

Sabot or heat rounds fired from the M1A1 SA tank travel in an arc, not a straight line. The calculation includes the mass of the projectile, air resistance, and the drag coefficient. VR-Engage calculates for collisions in each frame, which can impact performance.

➤ **To fire a ground vehicle’s gun, do one of the following:**

- Press the right trigger on your game controller.
- Press **F**.
- Click the left mouse button.

6.3.2 Choosing a Weapon

You can cycle through available munitions to fire the one you want.

➤ To cycle through the weapon list using a game controller, use the left trigger.

➤ To select a weapon using the keyboard, press **R** for the next weapon or **V** for the previous weapon.

➤ **To cycle weapons using the Action menu (LAV only):**

1. Open the Action menu.
2. Use an Action menu control method to use the **Cycle Weapon** option.

6.3.3 Moving the Turret

The gunner and gunner/driver roles can move the turret on vehicles that have one.

➤ To move the turret in the gunner role, use the right thumbstick on your game controller. You can also use the mouse.

➤ To move the turret in the driver/gunner role, use the D-pad.

➤ To move the turret in either role, press the **H**, **K**, **U**, and **J** keys.

6.3.4 Stabilizing a Vehicle's Gun

By default, the LAV, Stryker, GTK Boxer, and the M1A1 and M1A2 tanks allow you to stabilize the guns. This locks the azimuth and elevation to the values they had with respect to the world when the turret was last moved.

If you have VR-Forces with the VR-Engage add-ons, you can configure whether gun stabilization is enabled by default by editing the weapon system in the Simulation Object Editor. See the information on editing a system's properties in the *VR-Forces Users Guide*.

➤ **To enable or disable gun stabilization:**

1. While acting as the gunner, open the Action menu.
2. Choose **Gun Stabilization**.
3. Use an Action menu control method to toggle the setting: **Heading** stabilizes the gun and **Disabled** does not stabilize the gun.

6.4. The Commander Roles

There are several types of commander roles in VR-Engage: Commander, Commander Hatch, and Commander RWS (GTK Boxer CRV only). Commanders can rotate the camera to look around, zoom in and out, change the sensor view, and trigger events. Commanders can also communicate with other crew members and other players using the intercom or radio. On some vehicles, the commander role is also configured to take over the gunner role, slew the gun to a target, and designate a target with a laser.

- To learn about taking over the gunner role, see "6.4.1 Overriding the Gunner" on the facing page.
- To learn how to slew to cue, see "6.4.2 Slewing the Gun to Designate a Target" on the facing page.
- To learn how to use the laser designator, see "6.4.3 Using the Laser Designator" on the facing page.
- To learn more about triggering events, see "4.4. Starting VR-Forces Events" on page 102.



The Commander RWS role on the GTK Boxer CRV can perform some gunner tasks as well as commander tasks.

6.4.1 Overriding the Gunner

On ground vehicles that are configured for it, the Commander RWS role can take over the gunner role controls to point the gunner at a target.

➤ **To override the gunner controls:**

1. While engaged in the Commander RWS role, open the Action menu.
2. Choose **Gunner Override**. The view switches to the gunner.
3. Move and fire the gun as needed. See "6.3. The Gunner Role" on page 139 for instructions.
4. When you are finished, open the Action menu and then choose **Exit Gunner Override**.



To enter or exit Gunner Override, you can also press **O**.

6.4.2 Slewing the Gun to Designate a Target

The Commander RWS role can designate a target for the gunner (slew to cue).

➤ **To slew the gunner's gun to a specific point:**

1. Use the controls to focus your eyepoint on the target.
2. Press **C**. You take temporary control of the gun, slewing it to point at the target.
3. Release the key or button to return control to the gunner.

6.4.3 Using the Laser Designator

On the GTK Boxer CRV, the Commander RWS role can use the laser designator to lase a target. To move the turret, you can use the right thumbstick on a game controller, mouse, or keyboard. See "6.3.3 Moving the Turret" on the previous page.

➤ **To use the laser designator,**

1. Point the turret at what you want to target.
2. Press the **L** key or the A button on the game controller.

➤ **To specify a laser code:**

1. Open the Action menu and then choose **Laser Code**.
2. Use the keyboard to enter a number and then press **Enter**.

➤ To stop lasing, open the Action menu and then toggle Lasing to Off.

6.5. Using the Intercom

Players who are embarked on a ground vehicle can communicate using the intercom. The intercom is voice activated and transmits any sound that is greater than the VOX threshold.

➤ **To use the intercom:**

1. Open the Action menu.
2. Choose **Radio**.
3. Cycle through the options to choose **Intercom**.
4. Start talking.

6.6. Changing the Sensor for a Vehicle Crew Member

The Gunner, Commander, and Commander RWS roles can change their sensor mode.

➤ To change the sensor for a gunner or commander RWS, use right thumbstick (press) until you see the sensor that you want.

➤ To toggle the sensor for a commander, press down on the D-pad down.

6.7. Pivoting a Tank

Tanks can pivot in place.

➤ **To pivot a tank:**

1. Start the engine.
2. Press the X button on the game controller or **C** on the keyboard. The Gear setting changes to Pivot in the HUD overlay.
3. Put the throttle forward to engage the engine.
4. Use the right thumbstick to pivot left or right.



You can also enable or disable Pivot by toggling the setting in the Action menu.

6.8. Emergency Lights and Sirens

The police cruiser and ambulance entities have emergency lights and sirens.

➤ **To turn on emergency lights and sirens:**

1. Open the Action menu.
2. Choose **Emergency Lights** or a specific type of siren.
3. Toggle the option on.

6.9. Configuring Ground Vehicles on Geocentric Terrains

VR-Engage ground vehicles use the Vortex physics libraries for their movement systems. When a vehicle moves on a geocentric terrain, VR-Engage must convert the geocentric coordinates to a flat earth coordinate system. The conversion is based on a user-defined origin that, ideally, represents the location where the majority of simulation activity will take place. As vehicles move further away from the origin,

projection errors can occur. When distances from the origin exceed several hundred kilometers, the projection error may be noticeable and vehicles may drift from their expected line of movement.

To resolve this problem, you should set the origin to a location that is close to the location where you are simulating vehicles. For example, by default, VR-Engage sets the origin to Honolulu, Hawaii, USA, the location of the terrain we use for the Test Drive and First Experience scenarios. If you want to simulate in some other location in the world, you should set the origin differently.

To set the origin, specify values for the **coordinateConverterOrigin** parameter in *./appData/scripts/vortexConfig.lua*. The values are in decimal degrees for latitude and longitude and meters for elevation. If the parameter is not specified, the default values are 0,0,0.

This parameter is only used for geocentric terrains. For projected terrains, the terrain data is used directly and no conversion is needed.



7. Controlling Aircraft

This chapter explains how to control aircraft.

7.1. Introduction 147
7.1.1 Aircraft Roles 147
7.1.2 The Head-Up Display 147
7.1.3 The Instrument Panel 149
7.1.4 Aircraft Display Options 151
7.2. Taking Control of an Aircraft 152
7.3. Using Autopilot 153
7.3.1 Setting the Autopilot Mode 153
7.4. Selecting a Bullseye 154
7.5. Using the Radar Warning Receiver 154
7.6. Controlling Landing Gear 155
7.7. Controlling Flaps 156
7.8. Launching Countermeasures (Chaff and Flare) 156
7.9. Choosing a Weapon 156
7.10. Air-to-Ground Combat 157
7.10.1 Targeting an Air-to-Ground Weapon Using MGRS Coordinates 157
7.10.2 Targeting an Air-to-Ground Weapon Using a Laser Code 158
7.11. Air-to-Air Combat 158
7.11.1 Selecting Targets 159
7.11.2 Firing an Air-to-Air Weapon 162
7.11.3 Focusing the Sensor on the Radar Target 162
7.12. Using the Common Operating Picture 162
7.12.1 Displaying COP Information for an Entity 163
7.12.2 Toggling the COP Focus Area 163
7.12.3 Selecting an Entity for the COP 164
7.13. Requesting SAR Images from RadarFX Server 164
7.13.1 Requesting a SAR Image 164

7.13.2 Configuring a SAR Request 165

7.1. Introduction

VR-Engage allows you to control fixed-wing aircraft and rotary-wing aircraft. VR-Engage uses the RTDynamics RotorLib and FixedWingLib libraries to provide realistic flight dynamics and aircraft controls. VR-Engage includes mappings for various controllers; see "11.8. Flight Controllers" on page 201 for details. (You can configure VR-Engage to work with other controllers. See "11.12. Configuring Games Controllers and Keyboard Control" on page 219.)

When you are flying an aircraft, you have an out-the-window head-up display (HUD) and an instrument panel. VR-Engage also supports a 3D cockpit. For more information about display options, see "7.1.4 Aircraft Display Options" on page 151.

For a list of aircraft included with VR-Engage, see "Chapter 12. Controllable Entities" on page 239. You can add more aircraft. For information, see "Chapter 13. Adding and Editing VR-Engage Entity Models" on page 275.

7.1.1 Aircraft Roles

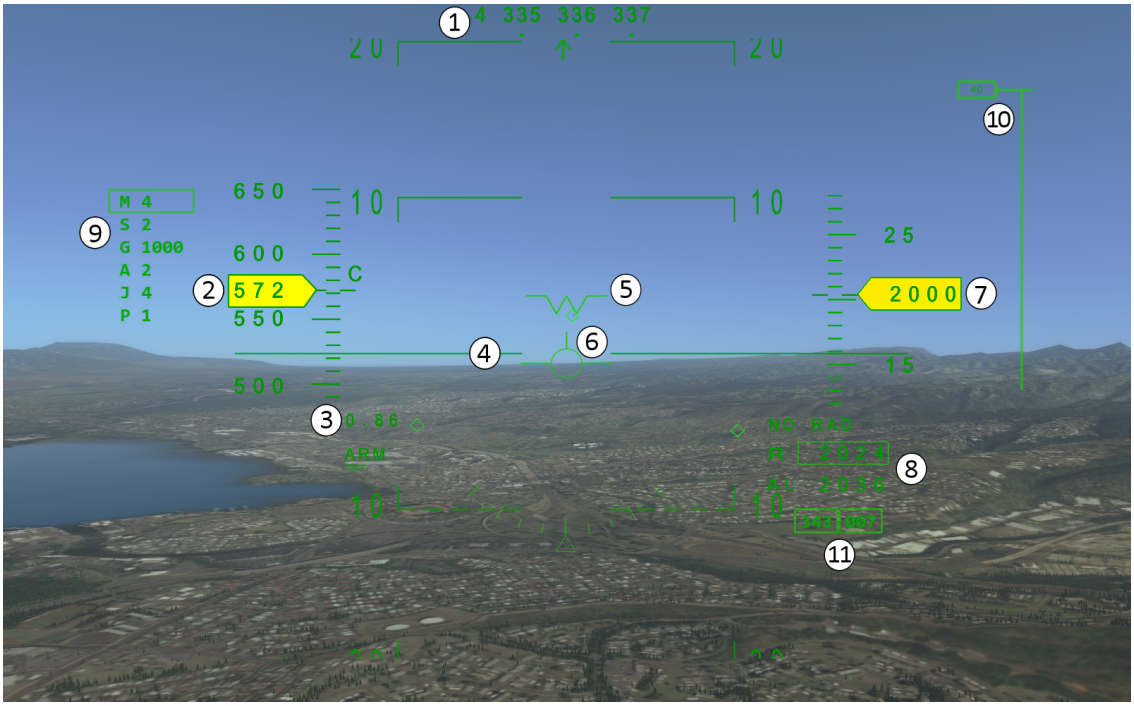
All piloted aircraft support the Pilot role, while UAVs support the Sensor Operator role. Many piloted aircraft also support the Sensor Operator role. Some aircraft support other roles that you can select from the Choose Role Panel.

To learn what roles are available for each aircraft, see "Chapter 12. Controllable Entities" on page 239.

7.1.2 The Head-Up Display

The head-up display (HUD) provides an out-the-window view with an overlay showing important flight instrument information.

Figure 54. HUD



The information shown includes:

- 1 – Heading Indicator
- 2 – Airspeed Indicator
- 3 – Mach Number (fixed wing only)
- 4 – Attitude Indicator (Artificial Horizon)
- 5 – Flight Path Indicator - direction aircraft is facing
- 6 – Flight Path Indicator - direction of aircraft's movement (fixed wing only)
- 7 – Altimeter
- 8 – Altitude (R = above ground level; AL = above mean sea level)

Aircraft entities with weapons display additional information:

- 9 – Weapon System Counts
- 10 – Launch Success Zone (for selected weapon system)
- 11 – Bearing and Distance to Bullseye

The weapon system counts display only the weapons configured for the entity. The list can include up to seven weapon systems.

The full acronyms or names of the weapon systems are shown in the Weapons and Stores multi-function display in the instrument panel (Figure 55 below) and the Action menu.

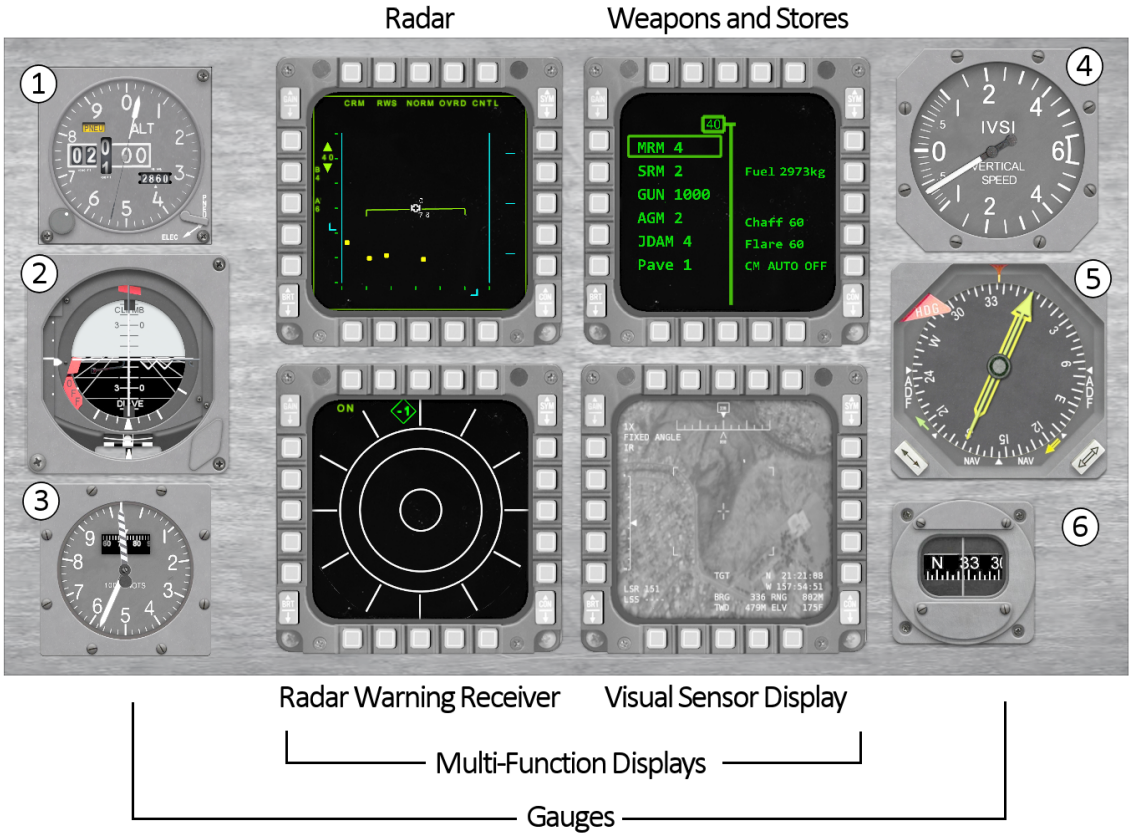
The simple Launch Success Zone shows only the weapon's minimum range, maximum range, and range to the currently selected target. The number indicates the weapon's maximum range in nautical miles.

7.1.3 The Instrument Panel

The Instruments Only display shows the instrument panel (Figure 55), which has six gauges and four multi-function displays (MFD). (The 1 Screen Horizontal display shows six gauges and two MFDs on the instrument panel.)

The instrument panel is an independent window. You can minimize and maximize it. If you are using one monitor, it may be hidden behind the HUD. If that happens, press **Alt**+**Tab** to bring it to the front.

Figure 55. Instrument panel



The gauges are:

- 1 – Altimeter
- 2 – Attitude Indicator (Artificial Horizon)
- 3 – Airspeed Indicator
- 4 – Vertical Speed Indicator
- 5 – Heading Indicator
- 6 – Compass

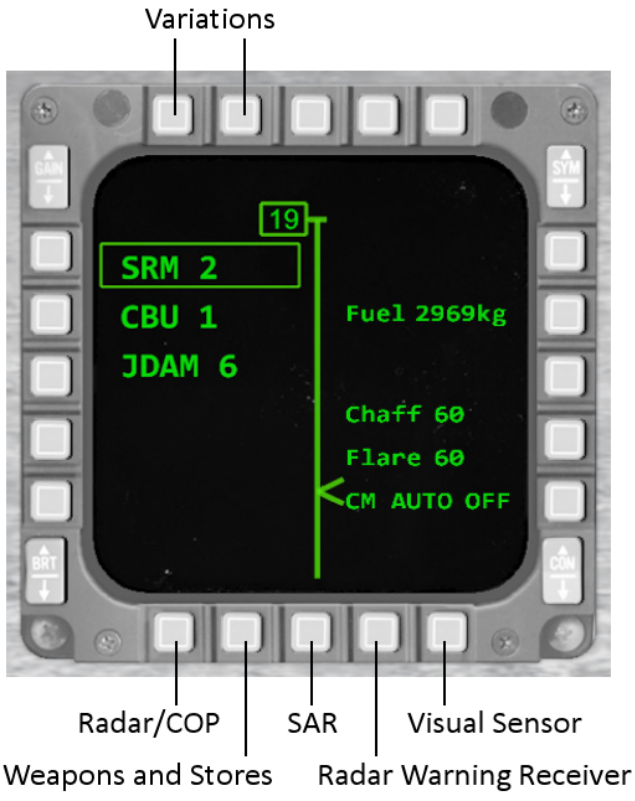
The MFDs are:

- Tactical Situation Display (radar or radar system).
- Weapons/Stores Control Panel.
- Synthetic Aperture Radar (SAR) view.
- Radar Warning Receiver. Shows entities that are detecting you on their radar. The rings show the effective threat range.
- Visual Sensor Display. Displays the view from the gimbaled sensor or a plan view display of the scenario.

Changing the Multi-Function Display

Each MFD window can display any of the possible display types.

Figure 56. Multi-function display



➤ To select a display, click one of the buttons along the bottom of the display.

➤ To view a variation of the display, use either of the first two buttons at the top of the display.

- If the instrument panel does not respond when you click a button, it might not have focus. Click it a few times or expand the window to gain focus.
- You cannot change the multi-function display in a 3D cockpit.

7.1.4 Aircraft Display Options

VR-Engage has the following options for displaying the head-up display (HUD) and instruments panel:

- 1 Screen Horizontal. HUD and instrument panel are integrated in one window, optimized for a single monitor in horizontal (landscape) orientation.
- 2 Screen. The HUD and instrument panel are in separate windows. If you have two monitors you can move the instrument panel into the second monitor to see both displays at the same time. If you choose this option and are using one monitor, the HUD is full screen and the instrument panel is in a separate window (which might be hidden behind the HUD).
- 2 MFD Panel. Displays the HUD and two MFDs.
- HUD Only. Displays the HUD only, optimized for a monitor in horizontal (landscape) mode.
- Instruments Only. Only the instrument panel displays.
- 3D Cockpit. A 3D cockpit that has the HUD and instruments (Figure 57). In this display option, you cannot change the multi-function display. The 3D cockpit is not available for all aircraft. (In VR-Engage 2.0.1, it is available for the Beechcraft MC-12 King Air, Beechcraft MC-12 King Air Single Seater, and T38-C Talon jet.)
- VR Headset. Select this option if you are using a head-mounted display. (Available only if an HMD is connected.)
- MR Headset. Select this option if you are using a mixed reality head-mounted display. That is, an HMD with a camera attached so that you can see both the virtual world and real world objects such as your hands and joysticks. (Available only if an HMD is connected.)

Figure 57. 3D cockpit



➤ **To choose an aircraft display option:**

1. In the Choose Role Panel, select the aircraft you want to control.
2. In the Display(s) list, select the display option.

If you experience problems with a display mode, make sure that vrEngage.exe is included in the BlackApps.dat file. See "A.1. Performance and Display Issues" on page 314.

7.2. Taking Control of an Aircraft

To take control of an aircraft, you select it in the Choose Role Panel, just as you would for a human character or a vehicle.

It is beyond the scope of this manual to teach you how to fly a fixed-wing or rotary-wing aircraft. For a brief tutorial on flying the aircraft in VR-Engage, see the section for fixed-wing aircraft in the chapter *Controlling Entities in VR-Engage* in *VR-Engage First Experience*.

You can use an HMD while controlling an aircraft. Hand controllers are not supported for flying aircraft. You must still control them using a game controller or flight controller.

If you take control of an aircraft that is in the air, you must start flying the aircraft as soon as you click Engage. Even though it may take several seconds or longer to display the HUD and instrument panel, the aircraft is already flying. If you take control of an aircraft that is close to the ground, such as a low flying helicopter, there is a risk that it could crash before the HUD is available if you do not immediately maintain its altitude.

7.3. Using Autopilot

You can set an aircraft to use autopilot while still otherwise controlling it. Autopilot has the following modes:

- ALT. Set the altitude and use the current heading and speed. (Equivalent to VR-Forces Fly Altitude task.)
- HDG. Set the heading and use the current altitude and speed. (Equivalent to VR-Forces Fly Heading task.)
- SPD. Set the speed and use the current heading and altitude.
- ALT/HDG/SPD. Fly at the specified altitude, heading, and speed. If no altitude, heading, speed, or any of those are specified, use the aircraft's current settings.
- PATH. Fly along a route that you select. (Equivalent to VR-Forces Move Along Route task.)



If a helicopter is on the ground and you set it to ALT/HDG/SPD without specifying an altitude, it takes off, moves to an altitude of 100 feet and flies at its current heading. If a fixed-wing aircraft is on the ground, it does not take off. For more information about how aircraft respond to ALT/HDG/SPD, please see the Fly Heading and Altitude task in the *VR-Forces Users Guide*.

If you are using a Thrustmaster HOTAS Warthog, pressing the Autopilot button when the toggle is set to PATH button causes the plane to fly a pattern hold around a waypoint called PATH, if it exists. Otherwise it flies a pattern hold around the current location.

7.3.1 Setting the Autopilot Mode

➤ **To set the autopilot mode or turn off autopilot:**

1. On the Action menu, choose **Autopilot**. The menu changes to the Autopilot menu.
2. Choose a setting. The Autopilot menu changes as follows based on the autopilot setting you choose:
   - If you choose **ALT**, the **Set Hold Altitude** option is added to the menu.
   - If you choose **HDG**, the **Set Hold Heading** option is added to the menu.
   - If you choose **SPD**, the **Set Hold Speed** option is added to the menu.
   - If you choose **ALT/HDG/SPD**, **Set Hold Altitude**, **Set Hold Heading** and the **Set Hold Speed** options are added to the menu.
   - If you choose **PATH**, you can select from a route in the scenario and the **Set Hold Speed** option is added to the Autopilot menu.
3. Choose the altitude, heading, or speed option that you want to set. An input dialog box opens.
4. Type the value that you want to set.
5. Press **Enter**.

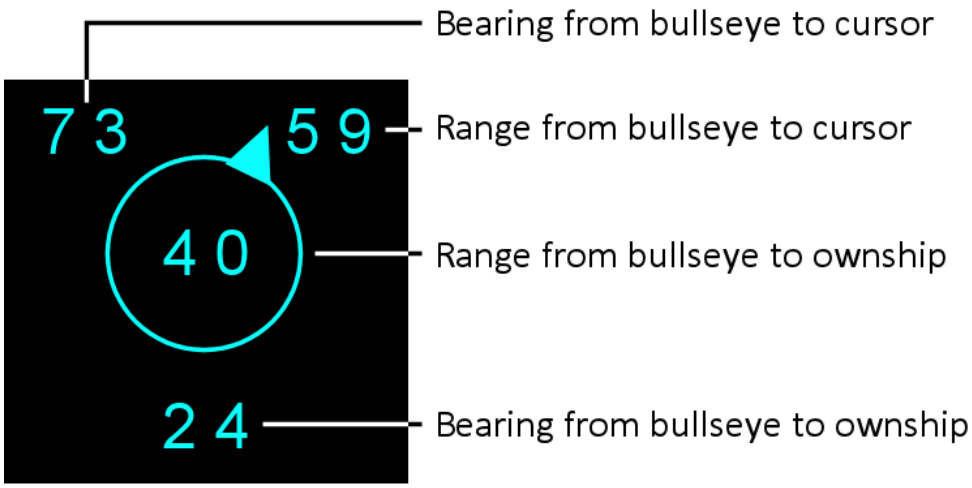
7.4. Selecting a Bullseye

Bullseyes are points that are specified before a pilot flies a mission. While the pilots are on the mission, they report their position—bearing and distance—relative to that bullseye. This allows the pilots to provide an accurate report of their location that opposing forces will not recognize.

If you do not specify a bullseye, the HUD displays the bearing and distance as 000 and 000. After you select a bullseye, the HUD displays the bearing and range of your position relative to the bullseye's coordinates (Figure 54 on page 148). The radar system displays the bearing and range from the bullseye to the cursor, and the bearing and range from the bullseye to the ownship. All ranges are measured in nautical miles, and distances are calculated using the great circle distance.

The bullseyes that you can select when engaged in a pilot role are tactical graphics added to a scenario in VR-Forces.

Figure 58. Bullseye display in radar system



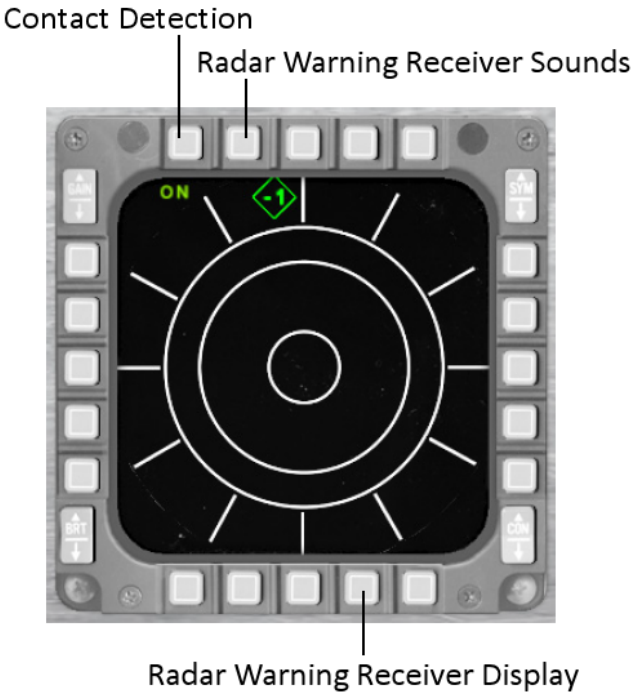
➤ **To set a bullseye:**

1. On the Action menu, choose **Select Bullseye**. The Bullseyes dialog box opens, listing the bullseye tactical graphics defined in the simulation.
2. Use an Action menu control method to select the desired bullseye.

7.5. Using the Radar Warning Receiver

Rotary-wing and fixed-wing entities have a radar warning receiver (RWR). It sounds a warning if it detects radar in addition to displaying the radar contacts. You can enable or disable the RWR and the sound.

Figure 59. Radar Warning Receiver



➤ **To enable or disable the radar warning receiver:**

1. In one of the multi-function displays (MFD) of the instrument panel, click the fourth button at the bottom to show the RWR.
2. Click the first button at the top of the MFD to turn contact detection on or off.

➤ **To enable or disable the radar warning receiver sounds:**

1. Open the Action menu.
2. Choose **RWR Sounds**.
3. Use an Action menu control method to toggle sounds on or off.

You can also toggle the radar warning receiver sounds by pressing **M** or clicking the second button at the top of the multi-function display.

7.6. Controlling Landing Gear

Fixed-wing entities can raise and lower landing gear.

➤ **To control landing gear:**

1. Open the Action menu.
2. Choose **Landing Gear**.
3. Use an Action menu control method to select Up or Down.

You can also toggle landing gear by pressing **G**.

7.7. Controlling Flaps

You can set the flaps on a fixed-wing entity up or down.

➤ **To control flaps:**

1. Open the Action menu.
2. Choose **Flaps**.
3. Use an Action menu control method to toggle the landing gear to Up or Down, MVR (Maneuver).

You can also control flaps using a joystick or game controller. See "11.8. Flight Controllers" on page 201 for details.

7.8. Launching Countermeasures (Chaff and Flare)

Aircraft configured with countermeasures have the ability to automatically launch chaff and flare when they detect a threat. You can also launch chaff and flare manually.

➤ **To enable or disable countermeasures:**

1. Open the Action menu.
2. Choose **Auto Launch CM**.
3. Use an Action menu control method to toggle automatic countermeasures on or off.

➤ To launch countermeasures manually, press **C** for chaff and **F** for flare.

7.9. Choosing a Weapon

Aircraft have specific loadouts of air-to-ground and air-to-air weapons, which display on the left side of the HUD. You can choose the weapon that you want to use. In the HUD (Figure 54 on page 148) and the Weapons and Stores display (Figure 56 on page 150), a rectangle indicates the selected weapon.

Table 4. Aircraft weapons

Weapon Type	HUD Abbreviation	Action Menu Abbreviation
Air-to-Air		
Long-Range Missiles	L	LRM
Medium-Range Missiles	M	MRM

Table 4. Aircraft weapons (continued)

Weapon Type	HUD Abbreviation	Action Menu Abbreviation
Short-Range Missiles	S	SRM
Air-to-Ground		
Gun	G	GUN
Bombs		
Cluster	C	CBU
JDAM	J	JDAM
KAB	K	KAB
Paveway	P	Pave
Air-to-Ground Missiles	A	AGM

➤ **To select a weapon:**

1. Open the Action menu.
2. Choose **Selected Weapon**.
3. Use an Action menu control method to cycle through the weapons available on your aircraft until the one you want to use is selected.



You can also cycle through the weapon list using the Select Weapon button on your game controller or joystick, or by pressing **A**.

7.10. Air-to-Ground Combat

You can fire air-to-ground bombs or missiles. By default, they try to hit wherever the sensor view is looking. You can specify a target using MGRS coordinates, or you can lase the target and look at the location being lased. Lasing can be done by a VR-Forces entity or a VR-Engage character configured with laser designator. For lased targets, you can continuously track what the laser is targeting.

7.10.1 Targeting an Air-to-Ground Weapon Using MGRS Coordinates

MGRS coordinates must specify a zone, a grid, and an x, y offset. The x and y offsets must be the same number of digits. For example, a code of 04QFJ190557 would be built from:

- Zone: 04Q
- Grid: FJ
- X offset: 190
- Y offset: 557

In VR-Engage, if you change the coordinate system to MGRS (Display Settings dialog box, Display Units Settings page), the 3D entity label for an entity (windowed mode, 3D view) displays its location in MGRS coordinates. You can then use this value to specify the coordinates for dropping a bomb. To get the MGRS coordinates for a tactical graphic, you can view the state data in VR-Forces.

For more information about MGRS coordinates, see relevant pages available on the internet.

➤ **To fire an air-to-ground weapon using MGRS coordinates:**

1. Choose an air-to-ground weapon. See "7.9. Choosing a Weapon" on page 156).
2. On the Action menu, choose **Gimbaled Sensor**. The Gimbaled Sensor menu displays.
3. Choose **Target MGRS Coordinate**. The Input MGRS Coordinate dialog box opens.
4. Type the coordinates you want to target and then press **Enter**. The gimbaled sensor centers on those coordinates.
5. Press the trigger to fire the weapon.

7.10.2 Targeting an Air-to-Ground Weapon Using a Laser Code

You need the laser code to target an air-to-ground weapon with that code. This might be from someone who has the information from the scenario, or another player that is lasing the target as part of the scenario.

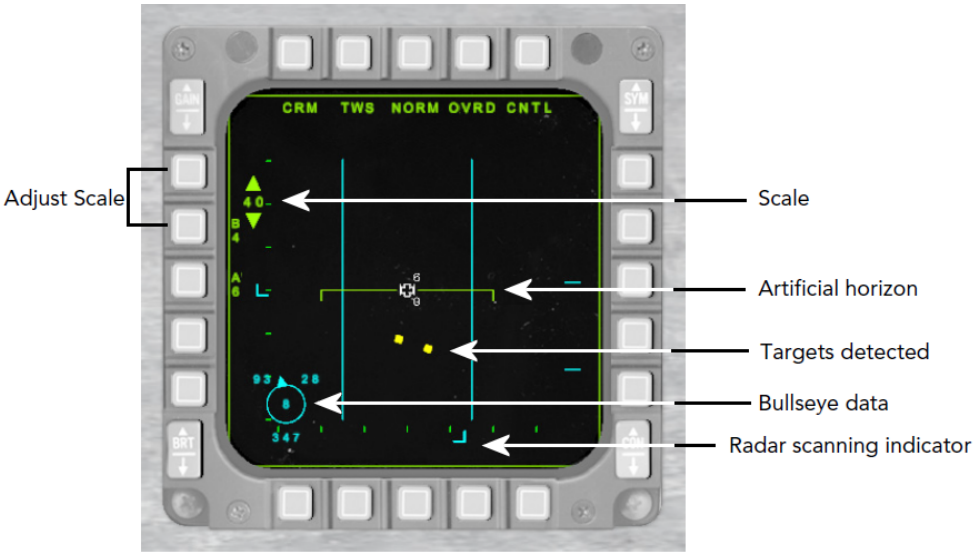
➤ **To target a laser spot:**

1. Choose an air-to-ground weapon.
2. On the Action menu, choose **Gimbaled Sensor**. The Gimbaled Sensor menu displays.
3. Choose **Target Laser Spot**. The Input Laser Code dialog box opens.
4. Type the laser code you are looking for and then press **Enter**. The gimbaled sensor centers on the lased target or location, and the LSS code shows that code and blinks in the Visual Sensor Display (if the display is active in your instrument panel).
5. Press the trigger to fire the weapon.

7.11. Air-to-Air Combat

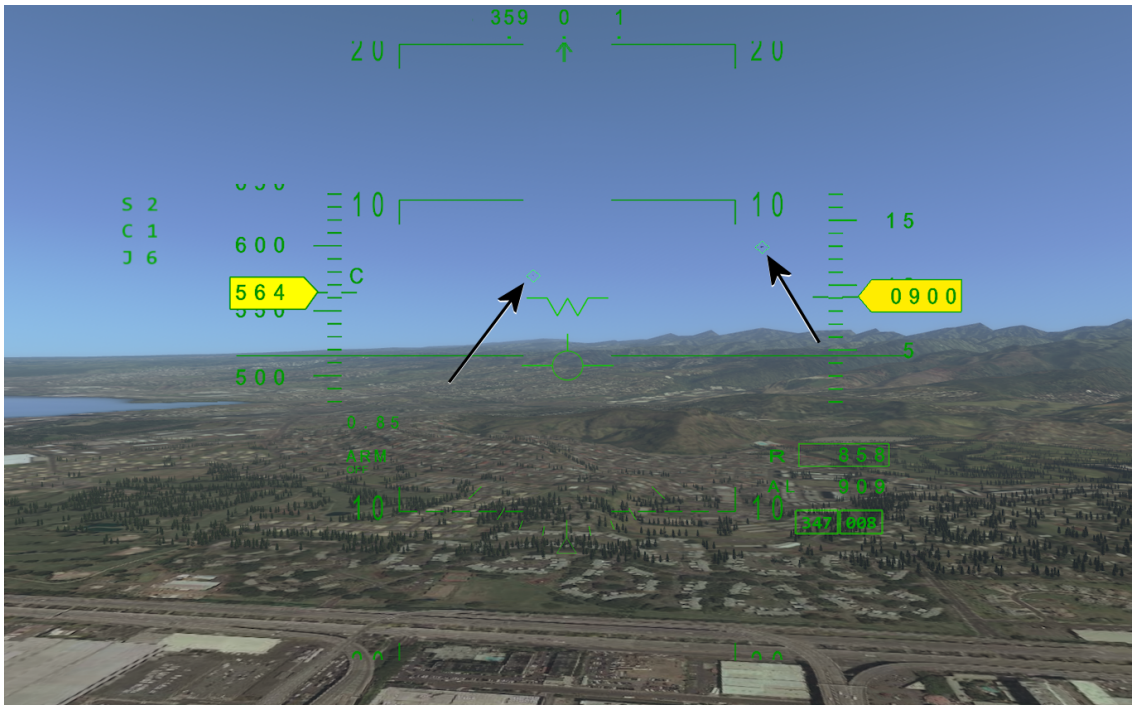
To fire a weapon at another aircraft, you must have a valid target in the Radar System (Figure 60). A blue cursor along the bottom of the Radar System shows when the radar is scanning. Yellow icons indicate detected targets. Detected contacts that are out of the selected range blink (increasing the range may shift their position and cancel the blinking).

Figure 60. Radar System



You can also see targets as green diamonds on the HUD (Figure 61).

Figure 61. Targets in HUD



7.11.1 Selecting Targets

Select a target using the cursor select controls on the joystick or keyboard. Once you have locked on a target (Figure 62), you can fire at it as long as it in range. See "Chapter 11. Mouse, Keypad, and Controller Mappings" on page 189 to learn about the cursor controls.

The Radar System has two modes: Range While Scan (RWS), which is the default setting, and Track While Scan (TWS). TWS, by default, has a narrower antenna sweep and collects more information (such as the target's velocity as well as its range). Both modes allow you to select multiple targets. To learn about targeting more than one entity, see "Selecting and Firing on Multiple Targets" on the facing page.

You can see that you have locked on a target when the following things happen in the Radar System:

- The first target icon you select becomes an outline and it is circled to indicate it is the current target. Each subsequent target icon changes to an outline.
- If you select the same target a second time, the radar switches to the Single Track Target (STT) sub-mode and you cannot select another target (other targets no longer appear in the display).
- If the radar is in STT mode, the antenna tracks the designated target instead of sweeping back and forth.
- The Track ID number for the target displays.
- The Track of Interest (TOI) readout line (ROL) displays for the selected target. If you move the cursor to another target, that target's information displays in the TOI ROL. If you have not selected the target, the text is black on white. Once you select the target, the text is white on black.

  The TOI ROL includes:

  - The Track ID
  - Target's bearing and range from bullseye
  - Target altitude
  - Target heading
  - Target speed (Mach number)
  - Target type
  - Target's bearing and range to target from your aircraft

The information for the currently selected target also displays in the HUD, and the selected target blinks (Figure 63).

Figure 62. Radar Lock in the Radar

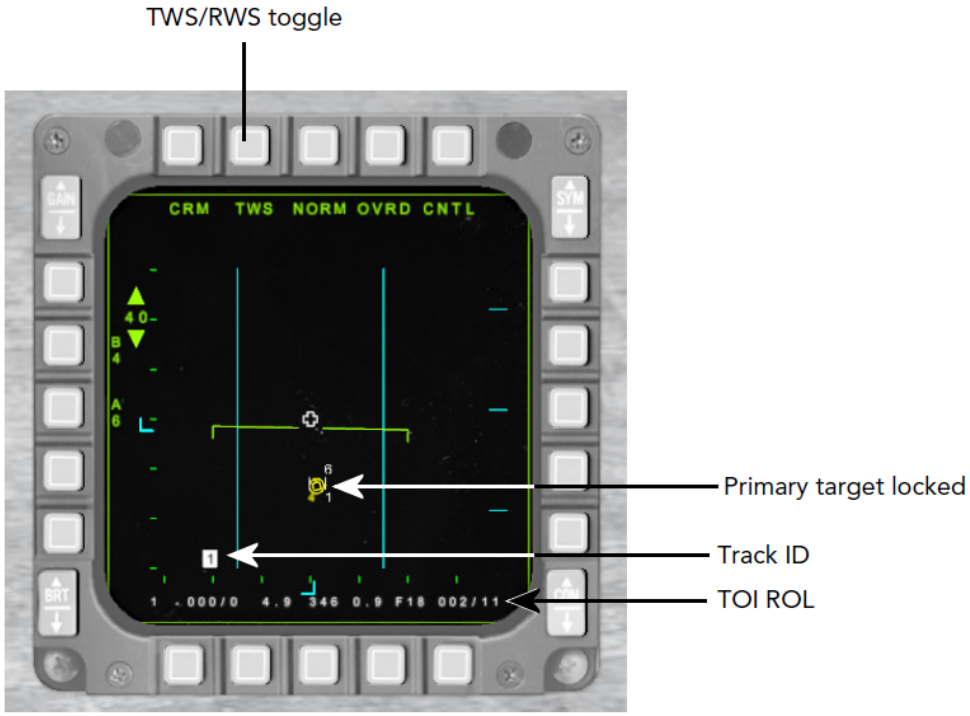
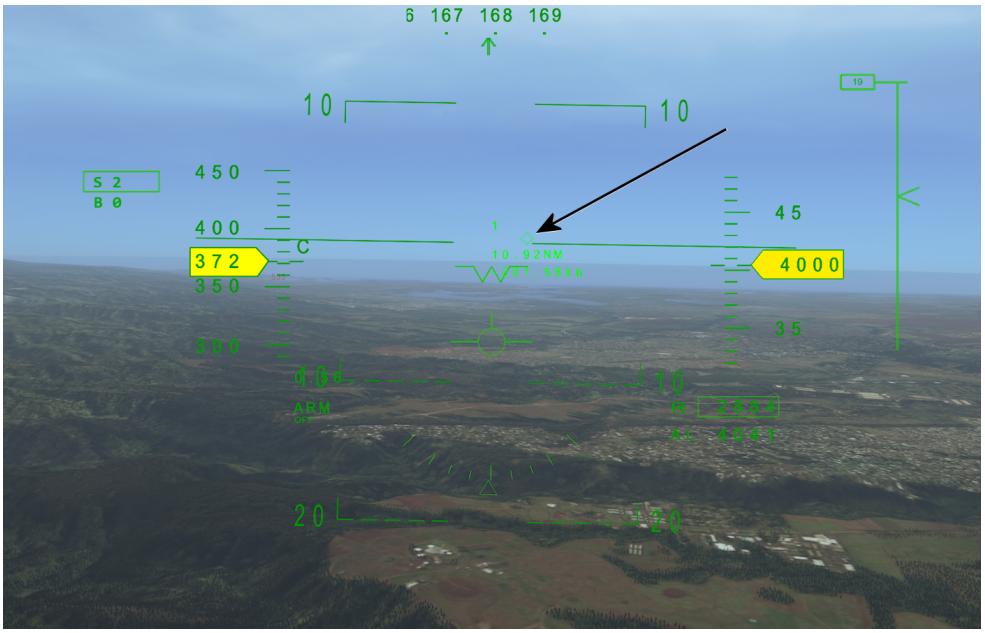


Figure 63. Radar Lock in HUD



Selecting and Firing on Multiple Targets

You can use the cursor controls in the Radar System to select multiple targets, up to six. The targets are added to an air target list in the order that you select them. This list determines the order of attack.

Each time you fire the selected air-to-air weapon, it fires at the next-to-shoot (NTS) target in the list and then cycles the list forward. The NTS target blinks in the HUD, and is circled in the radar. You can also cycle the list forward or backward using the cycling controls. For example, if you have targets 4, 5, 6, and 7, the weapon fires at the targets in order. When you fire the weapon or cycle the list forward, the order of targets becomes 5, 6, 7, and 4. If you cycle the list backward, the order is 7, 4, 5, and 6.

Targets are included in the list until they are destroyed, you unselect a target, or you clear the entire list. Additionally, changing the Radar System mode clears any selected targets and the list. You unselect a target or clear the air target list using the controls on the joystick or keyboard.

7.11.2 Firing an Air-to-Air Weapon

Air-to-air weapons use the radar for targeting.

➤ **To fire an air-to-air weapon:**

1. Choose an air-to-air weapon. See "7.9. Choosing a Weapon" on page 156.
2. Select the target. See "7.11.1 Selecting Targets" on page 159.
3. Press the right trigger to fire the weapon.

7.11.3 Focusing the Sensor on the Radar Target

You can set the gimbaled sensor to follow the current radar target.

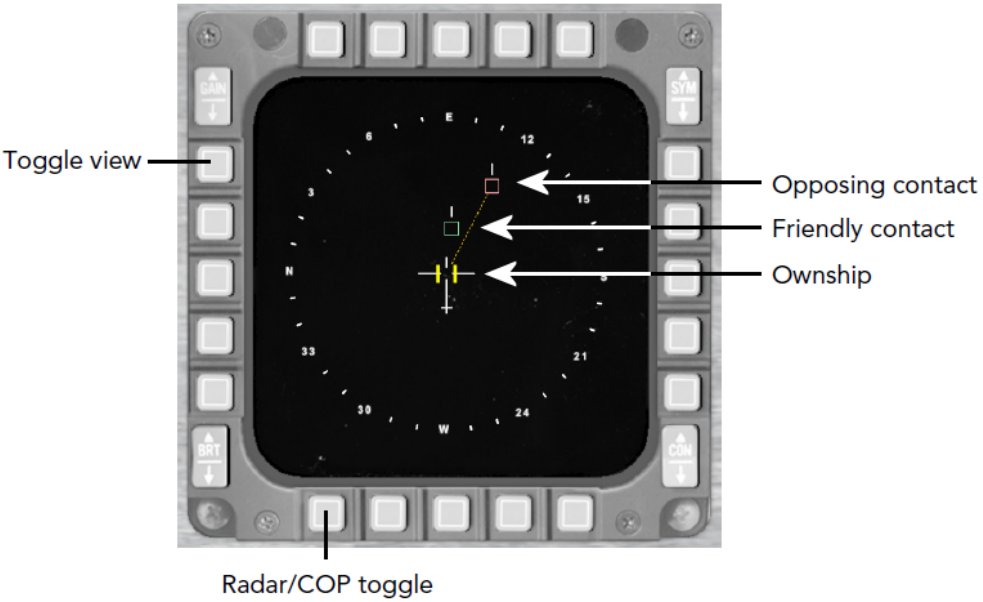
➤ **To focus the sensor on the radar target:**

1. First, target an aircraft. See "7.11.1 Selecting Targets" on page 159.
2. On the Action menu, choose **Gimbaled Sensor > Target Radar Target**. The Visual Sensor focuses on and follows the currently selected target.

7.12. Using the Common Operating Picture

The Tactical Data Link (TDL) Common Operating Picture (COP) is an alternate display to the radar system, displaying surveillance and TDL tracks. Only entities within range of your radar are displayed. In the Radar System, you can use the cursor to select which entity's engagement information to display.

Figure 64. TDL Common Operating Picture



➤ To view the TDL COP, in one of the multi-function displays (MFD) of the instrument panel, click the first button at the bottom to toggle between the Radar System and the COP.

7.12.1 Displaying COP Information for an Entity

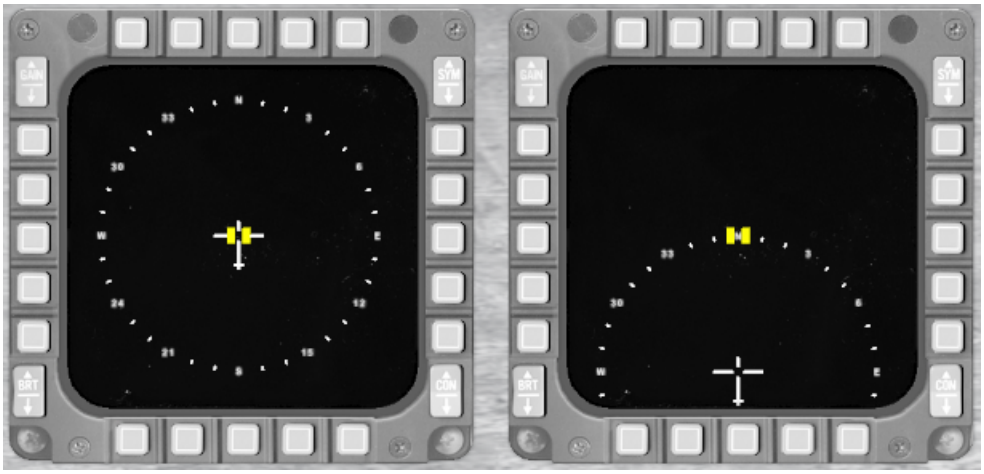
You can see the information about an entity detected by your radar system

➤ To display the information for an entity in the COP, use the controls on the joystick or keyboard to move the cursor over that entity. See "Chapter 11. Mouse, Keypad, and Controller Mappings" on page 189 to learn about the cursor controls.

7.12.2 Toggling the COP Focus Area

By default, the COP centers on the ownship entity. You can shift the focus to a point in front of the entity.

Figure 65. COP Focus Areas - Center and Forward



➤ To toggle the focus between centering on the ownship entity or a point in front of the entity, click the top button on the left side of the MFD.

7.12.3 Selecting an Entity for the COP

You can use the COP to display engagement lines between the aircraft you are flying and entities that you select with the radar.

➤ To select an entity, use the Radar System. See "7.11.1 Selecting Targets" on page 159.

7.13. Requesting SAR Images from RadarFX Server

VR-Engage can request synthetic aperture radar (SAR) images from RadarFX Server. To use this feature, you must install a licensed version of RadarFX Server. For details about using RadarFX Server, please see *RadarFX Users Guide*.

When you request an image, the server returns the image and it is displayed in the multi-function display window of an aircraft's instrument panel.

7.13.1 Requesting a SAR Image

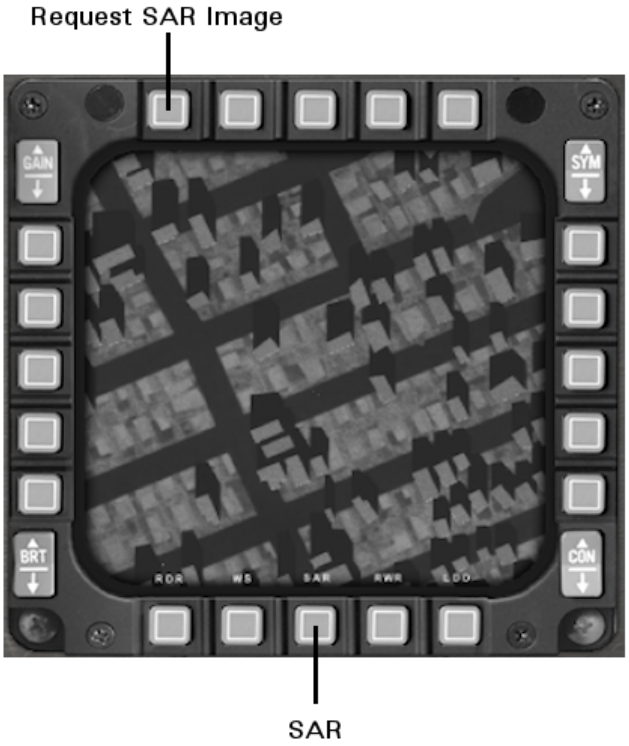
When you request a SAR image, VR-Engage sends the request using the settings configured in the SAR Request Settings dialog box.

RadarFX Server must be running when you request a SAR image.

➤ **To request a SAR image, do one of the following:**

- On the Action menu, choose **Request SAR Image**.
- On the multi-function display, while the SAR image is selected, click the Request SAR Image button (Figure 66).

Figure 66. SAR image



- Press the appropriate button on your joystick.
- Press **S**.

7.13.2 Configuring a SAR Request

When you request a SAR image, you must specify the amount power the SAR system should use. You can also configure the properties of the image that is returned.

When you request a SAR image as a pilot, the location of the sensor and area of interest are set for you. The image comes from the entity you're flying and focuses on the center of the gimbal view.

> You must set VR-Engage to Admin mode if you want to adjust the SAR Settings. See "3.1.3 Admin, User, and Lockdown Modes" on page 51.

Specifying the Power

The power value is the amount of power that the SAR system should use to generate the image. Higher power levels increase the image gain. Lower levels result in a darker image.

➤ **To specify the power**

1. Choose **Settings > SAR Settings**. The SAR Request Settings dialog box opens.
2. Select the Settings tab.
3. In the System Parameters group box, enter a value in the Power box.

Configuring Image Properties

You can specify the properties of the image that the server returns. You can also specify that the image be saved on the server. You can specify the following properties:

- Image size, in pixels.
- Image Rotation. The image can be rotated so that North is up or the sensor's view is up.
- Image Format. Options are PNG, JPG, BMP, Raw, and NITF (National Imagery Transmission Format).
- Image Data Type. Byte, short, or float.

➤ **To specify image properties:**

1. Choose **Settings > SAR Settings**. The SAR Request Settings dialog box opens.
2. Select the Settings tab.
3. In the Image group box, change properties as desired.
4. To save the image on the server, in the Save Options group box, select Save On Server. Specify the directory in which to save images.

Configuring the SAR Connection

To connect to a RadarFX Server, you must specify an IP address and port that matches the port used by the server. The default connection settings on the Connection tab of the SAR Request Settings dialog box match the default settings used by RadarFX Server. If you do not change the settings on the server, you do not have to make any changes in VR-Engage.



8. Controlling Drones

This chapter explains how to deploy and control drones in VR-Engage.

8.1. Introduction 168
8.1.1 Drone Camera Display 168
8.1.2 Video and Images 169
8.1.3 Drone Battery Life 169
8.2. Operating the Drone 169
8.2.1 Deploying and Flying the Drone 170
8.2.2 Stopping and Resuming Use of the Drone 170
8.2.3 Switching Control Between Character and Drone 170
8.2.4 Landing the Drone 171
8.2.5 Returning the Drone to its Starting Position 171
8.2.6 Recovering the Drone 171

8.1. Introduction

Drone operation is a feature of specific human entities in VR-Engage. It allows you to deploy and fly a quadcopter while playing the role of a human in the scenario.

The three entities that can deploy the quadcopter drone are:

- Civilian Female With Drone.
- Civilian Male With Drone.
- Police Officer With Drone.

After you deploy the drone, it appears and the perspective switches from first person to third person (Figure 67). It also displays the drone's camera view (Figure 68 on the facing page).

Your game controller controls the drone. You can use the **W**, **A**, **S**, and **D** keys to move the human entity. As the drone moves, the human character operating it pivots to keep it in the center of the main screen.

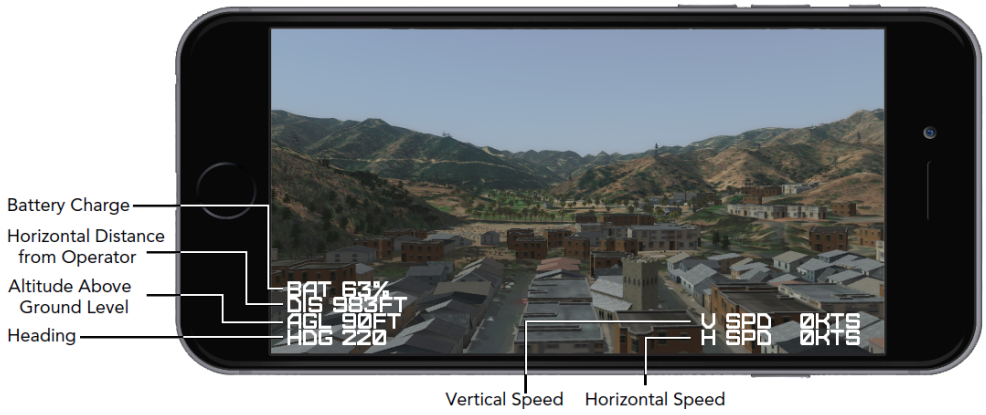
Figure 67. Civilian male with drone and drone camera view



8.1.1 Drone Camera Display

The drone camera view shows what the drone can see and also indicates the status of the drone (Figure 68).

Figure 68. Drone camera view and status indicators



The drone operator can pan the camera up or down, and also zoom in or out.

➤ **To operate the drone's camera:**

1. Add or select a character with a drone.
2. Engage the character.
3. To pan the camera up and down, use the game controller's D-pad (up/down).
4. To zoom in or out, use the game controller's right shoulder button or left shoulder button.

➤ To change the camera view in the interface, press **F6** to cycle the views.

8.1.2 Video and Images

You can use the drone to stream video and capture images. See "Chapter 10. Video Stream and Image Capture" on page 185.

8.1.3 Drone Battery Life

The drone has a limited battery life. The current charge status displays in the drone's camera view (Figure 68 above). If the battery charge falls below about 5% while it is in the air, the drone descends to the ground.

8.2. Operating the Drone

To deploy and operate a quadcopter drone, you must engage as one of the three entities with the drone (Civilian Female With Drone, Civilian Male With Drone, or Police Officer With Drone). The MAK One Demo scenario includes a Civilian Male With Drone entity in the neutral (green) force.

> You can create one of the drone operator entities to use if you do not want to use the MAK Suite Demo scenario. To learn how to create entities, see "3.2.2 Creating a New Entity" on page 54.

8.2.1 Deploying and Flying the Drone

When you deploy the drone, the default is to control its movements using a game controller and the character's movements with the keyboard.

➤ **To deploy the drone:**

1. Add or select a character with a drone.
2. Engage the character.
3. Open the Action menu and choose **Deployable Objects > Deploy Spark**.
4. Use the game controller to move the drone (see "11.4. Game Controller Mappings for Drones" on page 197). Use the **W**, **A**, **S**, and **D** keys to move the character.

8.2.2 Stopping and Resuming Use of the Drone

While the drone is deployed, you can stop or resume use of the drone.

➤ **To stop or restart using the drone:**

1. Open the Action menu and choose **Drone** to open the Drone submenu.
2. Choose the appropriate option:
   - **Stop Using Drone**. Closes the drone's camera view and leaves the drone in its current location. You can control the human character with the game controller.
   - **Use Drone**. Resumes using the drone. The camera view reopens.

8.2.3 Switching Control Between Character and Drone

You can change whether the game controller controls the human character or the drone.

➤ **To change what the game controller controls:**

1. Open the Action menu and choose **Drone**.
2. Use an Action menu control to toggle the **Gamepad Control**:
   - Human: The game controller controls the human character.
   - Drone: The game controller controls the drone.

You can also choose whether the human character's focus is on the drone.

➤ **To change the character's focus:**

1. Open the Action menu and choose **Drone**.
2. Use an Action menu control to toggle the **View Focus**:
   - Human: The human character no longer keeps the drone in the center of the screen.
   - Drone: The human character keeps the drone in his or her sights.

8.2.4 Landing the Drone

You can tell the drone to land itself at its current location.

➤ To land the drone, open the Action menu and choose **Drone > Land**. It descends to the ground.

8.2.5 Returning the Drone to its Starting Position

You can send a command to the drone to have it return from where it is currently located to the starting point where it was first deployed.

➤ To return the drone to its starting location, open the Action menu and choose **Drone > Return to Home**.

The drone camera view closes and the drone flies itself back to its starting position.

8.2.6 Recovering the Drone

When you are finished using the drone, you can command it to return and stow itself.

➤ To recover the drone, open the Action menu and choose **Drone > Recover**.

The drone camera view closes and the drone flies itself to your location from where it was. Afterward, its battery is immediately recharged to full.



If the drone is low on battery power, it cannot fly itself to you. You can move to the drone's location on the ground and then recover it.



9. Sensor Operator Role

This chapter explains the sensor operator role in VR-Engage. The sensor operator is included in the full version of VR-Engage and is also available as an independently licensed capability.

9.1. The Sensor Operator Role 174
- 9.1.1 Focusing the Sensor 175
- 9.1.2 Changing the Sensor Mode 176
- 9.1.3 Resetting the Sensor 176
- 9.1.4 Video and Pictures 176
- 9.1.5 Defining a Laser Code 176
- 9.1.6 Lasing Targets 177
- 9.1.7 Firing Weapons 177

9.2. Creating a Remote Camera Entity 177
9.3. Placing Sensors on Remote Entities 178
- 9.3.1 Configuring the Location of the Remote Camera 178

9.4. Configuring the Sensor Operator Overlay 181

9.1. The Sensor Operator Role

VR-Engage players can control sensors and remote cameras using the sensor operator role. The sensor operator can slew the sensor, zoom in and out, and can focus it on specific locations and entities.

Figure 69. Sensor operator view



You can add remote cameras to a scenario and set their heading and pitch. VR-Engage is also configured to automatically add cameras to remotely simulated (that is, non-VR-Forces) F-18 and UAV entities.

You can expand the remote cameras to other entities by modifying the componentAttachmentTable.mtl in the SMS.

If a sensor is mounted on an entity, the sensor operator does not control the entity. Therefore, you need another player to operate the entity, it must have a VR-Forces plan or task, or it must be mounted on a remote entity.

➤ **To operate a camera:**

1. In the Choose Role Panel, select a camera or an entity whose sensor you want to control.
2. In the Role list, choose Sensor Operator.
3. Click Engage. The window changes to 3D mode and a sensor overlay displays.
4. To zoom in and out, use the game controller's right thumbstick, press **Page Up** or **Page Down**, or use your joystick.
5. To slew the sensor, use the game controller's left thumbstick, press the arrow keys, or use your joystick.

9.1.1 Focusing the Sensor

You can focus the sensor on:

- The center of the screen.
- On a location specified by an MGRS coordinate or laser designation code.

  When you specify an MGRS coordinate or laser designation code, the sensor focuses on that spot. However, if a laser spot moves, the sensor does not follow it.
- On a waypoint. If a scenario has waypoints, you can focus on them.
- On a street. If a terrain has shapefiles that have street names, you can focus on a street. The sensor tracks the closest vertex. It must be within five kilometers of the X,Y coordinates of the sensor.

➤ **To set an MGRS target or laser code:**

1. Engage an aircraft as the sensor operator or engage a remote camera.
2. On the Action menu, choose **Gimbaled Sensor**. The Gimbaled Sensor menu opens.
3. Choose **Target MGRS Coordinate** or **Target Laser Spot**. A data entry dialog box opens.
4. Type the code.
5. Press **Enter**.

➤ **To track a waypoint:**

1. Engage an aircraft as the sensor operator or engage a remote camera.
2. On the Action menu, choose **Gimbaled Sensor**.
3. On the Gimbaled Sensor menu, choose **Track Waypoint**. A list of waypoints opens.
4. Use an Action menu control method to select the waypoint you want to track.

➤ **To track a street:**

1. Engage an aircraft as the sensor operator or engage a remote camera.
2. On the Action menu, choose **Gimbaled Sensor**.
3. On the Gimbaled Sensor menu, choose **Enter Street Name**. A text entry dialog box opens.
4. Type the name of the street you want to track.
5. Press **Enter**.

➤ To focus the sensor on the location currently at the center of the screen, press **L** or the A button on your game controller.

9.1.2 Changing the Sensor Mode

A sensor operator can view the scenario in visual mode, NVG mode, or IR mode.

You can set the available modes in *./data/simulationModelSets/VREngage/roles/sensorOperator.lua* by editing the **sensorModes** array. When you change the sensor mode, it cycles through the available modes in the order specified in the array.

➤ **To change the sensor mode:**

1. Engage an aircraft as the sensor operator or engage a remote camera.
2. On the Action menu, choose Gimbaled Sensor.
3. On the Gimbaled Sensor menu, choose **Mode**.
4. Cycle through the modes to choose the one you want.



You can also change the sensor mode by pressing **M** or the X button on a game controller.

9.1.3 Resetting the Sensor

When you reset the sensor, it returns to a fixed angle at the neutral azimuth and elevation.

➤ **To reset the sensor:**

1. On the Action menu, choose **Gimbaled Sensor**.
2. On the Gimbaled Sensor menu, choose **Reset Sensor**.



You can also reset the sensor by pressing **R** or the Y button on a game controller.

9.1.4 Video and Pictures

You can use the sensor operator to stream video and capture images. See "Chapter 10. Video Stream and Image Capture" on page 185 for details.

9.1.5 Defining a Laser Code

You can define the laser code used when you lase a target. Laser codes can be up to four digits long. See "9.1.6 Lasing Targets" on the facing page to learn how to lase a target.

➤ **To define the laser code:**

1. On the Action menu, choose **Set Laser Code**. Input Laser Code dialog box opens.
2. Enter a laser code and then press **Enter**. The LSR code displays in the sensor view.

Your laser designator uses that code.

9.1.6 Lasing Targets

Sensor operators can lase targets. When you lase a target, the display shows that the laser is on. If you enable laser designators, a green circle and laser code displays on the lased location. See "5.15.3 Enabling Laser Designators" on page 133.

➤ **To lase a target or turn off lasing:**

1. Focus the sensor on a target or location. See "9.1.1 Focusing the Sensor" on page 175.
2. On the Action menu, choose **Lasing**.
3. Use an Action menu control method to toggle Lasing on or off.

9.1.7 Firing Weapons

When you engaged as a sensor operator, you can use the weapons available on the aircraft. The typical case is to fire at your own laser designator. You can also target a laser spot that another entity is currently lasing.

➤ **To target, select, and fire a weapon:**

1. To fire at your own laser designator:
   a. Define your laser code (see "9.1.5 Defining a Laser Code" on the previous page).
   b. Focus the sensor and lase the target (see "9.1.6 Lasing Targets" above).

   To fire at another entity's laser designator, choose **Gimbaled Sensor > Target Laser Spot** and then enter that laser code. The LSS code displays in the sensor view.
2. In the Action menu, choose **Selected Weapon** and then use an Action menu control method to cycle through weapons until the one you want is displayed (Table 4 on page 156).
3. In the Action menu, choose **Fire Selected Weapon**.

> You can also press **W** to cycle the weapons, and **F** or the right trigger to fire the weapon.

9.2. Creating a Remote Camera Entity

When you host a session or run an unhosted session, you can add a remote camera (or other cultural feature), and attach it to (embark on) an entity.

➤ **To create the remote camera:**

1. In the Choose Role Panel, select the Create a New Entity option. The window lists the entity types that you can create.
2. Optionally, use the Categories list to filter for Cultural Feature.
3. Optionally, switch from the 3D view to the Plan View display mode (2D) by selecting an option at the bottom of the panel.

4. Select the Remote Camera entity.
5. Optionally, in the Marking box, you can type a name for the character. The maximum length is 11 characters.
6. Optionally, specify a heading. For remote cameras, the heading is relative to the object it is attached to.
7. Optionally, specify the initial pitch.
8. Click on the terrain or entity where you want to place the camera. (You can navigate the Plan View or 3D view using your mouse and keyboard. For details about navigating the terrain, see the chapter *Moving the Observer* in the *VR-Vantage Users Guide*. If you click on an entity, this attaches the camera to the entity.
9. If you simply want to observe the simulation from the perspective of the selected character but not actually control it, select the As Spectator check box.
10. If you want to use VR-Engage in full-screen mode, select the Full Screen check box.

    If you want to use VR-Engage in windowed mode, clear the Full Screen check box.
11. Click Engage. You can now control your character and participate in the scenario (assuming you did not choose to be a spectator).

9.3. Placing Sensors on Remote Entities

VR-Engage uses the VR-Forces remote component attachment feature to automatically place gimbaled sensors on remotely simulated entities. A remotely simulated entity is an entity that is part of an exercise, but is simulated by an application other than VR-Forces. The sensor is attached to the remote entity and moves with it. You can use the sensor operator role to operate the sensor. You cannot operate the remote entity.

The entities to which VR-Engage automatically attaches sensors are configured in *./data/simulationModelSets/VR-Engage/extra/componentAttachmentTable.mtl*.

The location where the sensor is embarked on the remote entity is configurable in the VR-Forces Simulation Object Editor.

If your scenario uses multiple simulation engines, then by default, the remote sensor is simulated on the first one.

9.3.1 Configuring the Location of the Remote Camera

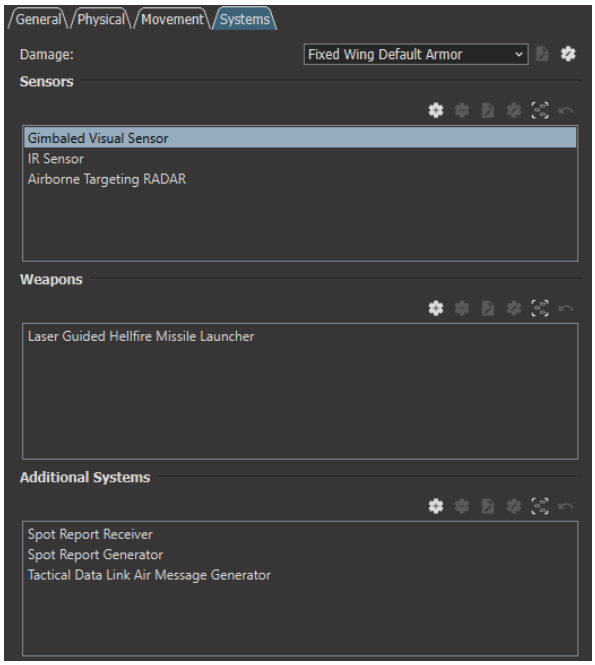
VR-Engage places a sensor on a remote entity at a location relative to the center of the entity. You can specify an offset from the default location.

➤ **To specify the offset for placing a sensor on a remote entity:**

1. In a version of VR-Forces that has the VR-Engage add-ons installed, open the Simulation Object Editor.
2. Open *VR-Engage.sms*.

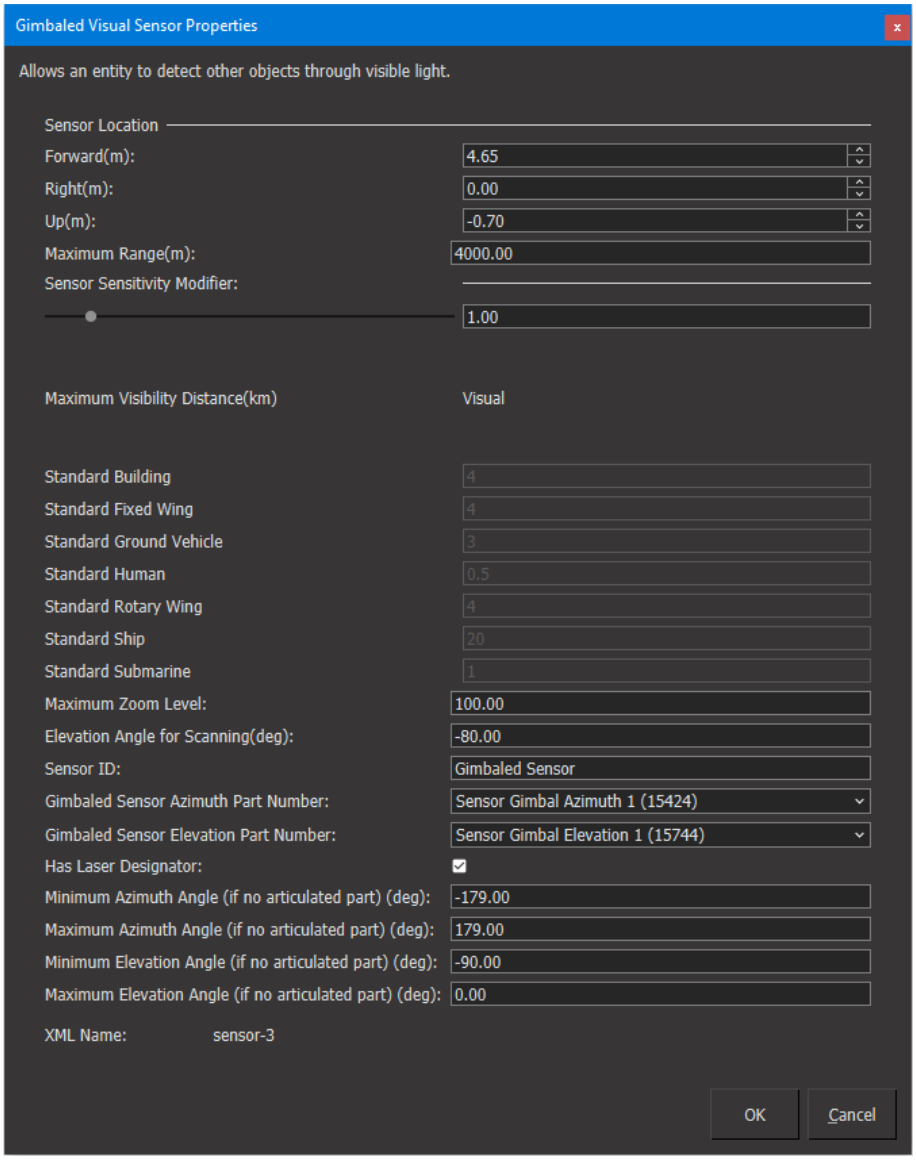
3. In the list of objects, select the VR-Forces entity that you want to configure to use a remote camera.
4. On the right, select the Systems tab.

Figure 70. Gimbaled Visual Sensor system



5. Select the Gimbaled Visual Sensor system.
6. Click the Properties button (). The Gimbaled Visual Sensor Properties dialog box opens.

Figure 71. Gimbaled Visual Sensor Properties



7. Edit the Sensor Location parameters.
8. Click OK.
9. Save the SMS.
10. Copy the updated SMS to your VR-Engage installation's *./data/simulationModelSets* directory. (Or you can just copy the *./data/simulationModelSets/VR-Engage/vrfSim/Remote Sensor.entity* file.)

9.4. Configuring the Sensor Operator Overlay

The information displayed on the sensor operator overlay is controlled by keywords in the following lines in *./data/simulationModelSets/VR-Engage/roles/sensorOperator.lua*:

```
...
  components = {
  ...
    ["vrvSensorOverlayUpdater"] = {
      --List of possible values for "sections":
      -- "Crosshair"         - Displays 4 corners of a box.
      -- "PlatformInfo"      - Displays information about ownship
entity.
      -- "VrePlatformInfo"   - Replaces "PlatformInfo". Adds Date and
Time displays.
      -- "TargetInfo"        - Displays ownship coordinates.
      -- "VreTargetInfo"     - Replaces "TargetInfo". Replaces
coordinate display with center of sensor's view.
      -- "VreMgrsTargetInfo" - Replaces "VreTargetInfo". Displays
coordinates in MGRS.
      -- "DisplayInfo"       - Displays sensor view information,
including mode, tracking mode and zoom level.
      -- "VreDisplayInfo"    - Replaces "DisplayInfo". Strips off
sensor mode "(CAMERAFX)" or "(SENSORFX)" suffixes.
      -- DisplayLaser"       - Displays Tracking laser code and
aircraft laser code.
      -- "Pitch"             - Displays ladder showing sensor elevation
relative to platform.
      -- "Compass"           - Displays ladder showing sensor azimuth
relative to platform, and "N" indicator displaying platform heading.
      -- "Reticle"           - Displays cross in the middle of the
screen.
      -- "DateTime"          - Displays date and time.
      -- "Time"              - Displays only time.
      -- "Rec"               - Displays static text showing "REC".
      componentType = "DtVrvSensorOverlayUpdater";
      priority = 10;
      --Defaults to VRV sections, with VRE updates for some sections.
      sections = {"Crosshair", "VrePlatformInfo", "VreTargetInfo",
"VreDisplayInfo", "LaserInfo", "Pitch", "Reticle", "Compass"};
    };
  };
  includes={
    "devices/radio.pilot.lua";
  };
```

The overlay to use is chosen based on the object type enumeration. You can add or remove overlay components by adding or removing keywords. The keywords are as follows, referenced to Figure 72 and Figure 73:

- Crosshair (1).
- VreTargetInfo (2). Show location using latitude/longitude coordinates.
- VreMgrsTargetInfo (2). Show location using MGRS coordinates.
- PlatformInfo (3).
- VreDisplayInfo (4).
- Pitch (5).
- Reticle (6).
- Compass (7).
- Time (8).
- LaserInfo (9). You define the Laser Code (LSR) for lasing a target by choosing **Set Laser Code** from the Action menu. You specify the Laser Spot (LSS) for the weapon by choosing **Gimbaled Sensor > Target Laser Spot** from the Action menu.
- Rec (Record. This is only text. VR-Engage does not record anything.) (10).
- DateTime (11).

Figure 72. Sensor operator overlay for aircraft

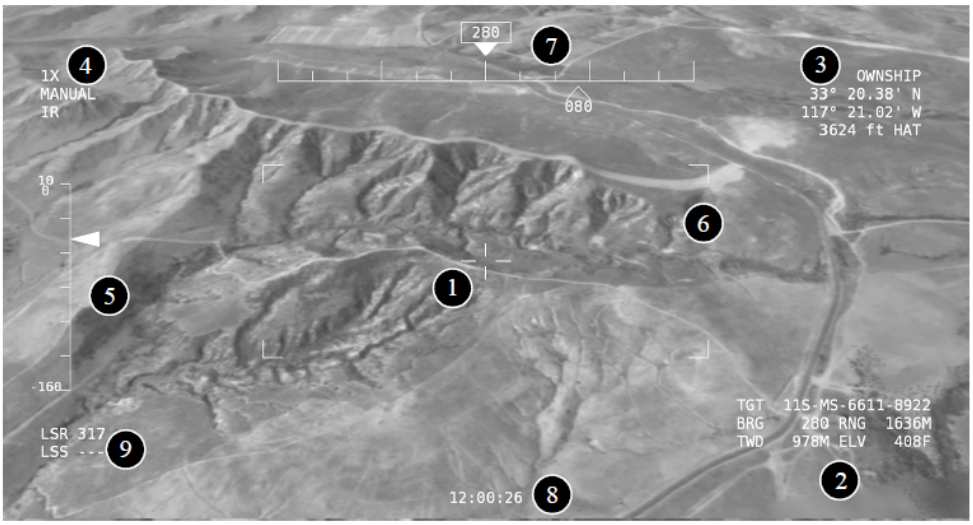
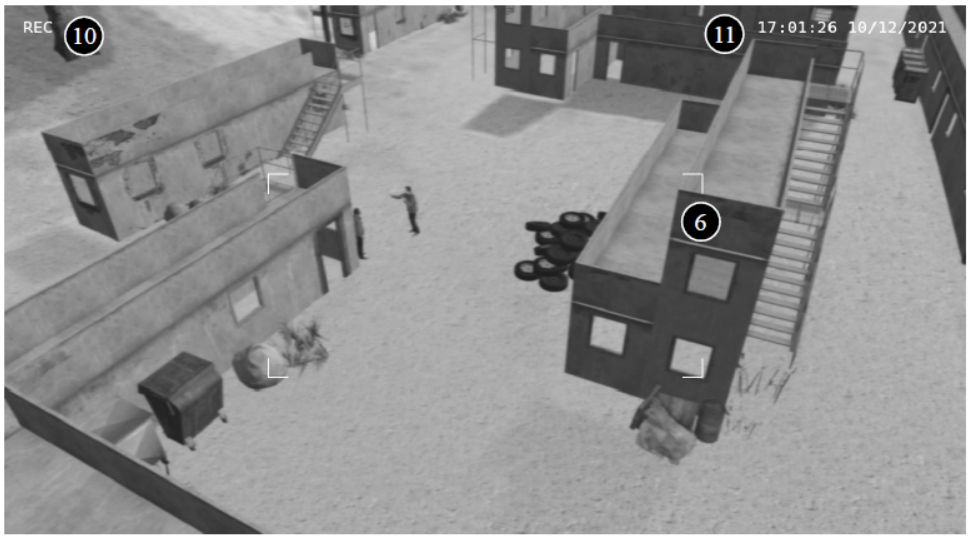


Figure 73. Sensor operator overlay for security camera



You can also write a plug-in to create new overlay elements.



10. Video Stream and Image Capture

This chapter explains how to use the Video Stream and Capture Image features for the Sensor Operator and Drone.

10.1. Introduction 186
10.2. Video Streams 186
- 10.2.1 Streaming Video as a Sensor Operator 186
- 10.2.2 Streaming Video from a Drone 186
- 10.2.3 Specifying the Video Stream File Settings 187
- 10.2.4 Viewing a Video File 187
- 10.2.5 Receiving a Video Stream 187

10.3. Capturing Images 188

10.1. Introduction

You can use the sensor operator or the drone operator roles to stream video and take pictures from their virtual sensors or cameras.



For more about those features, see "Chapter 8. Controlling Drones" on page 167 and "Chapter 9. Sensor Operator Role" on page 173.

10.2. Video Streams

Sensor operators and drones can stream video. The default is to "stream" (save) to a file, but you can also stream over a network. By default, video captures are saved to files in *./userData/videoCaptures*. See "10.2.3 Specifying the Video Stream File Settings" on the facing page for information on how to configure VR-Engage to stream video files to files in a different folder or to a network.

To play the video file or receive the video stream, you must install a media player that can process the video stream, such as VLC media player.

10.2.1 Streaming Video as a Sensor Operator

The sensor operator streams the camera's video feed.

➤ **To stream video as a sensor operator:**

1. Add or select an entity that has a sensor.
2. Engage as the Sensor Operator.
3. Open the Action menu and choose **Gimbaled Sensor**.
4. Use an Action menu control to toggle **Video Stream** to Started.

   A user can then view the file or receive the video stream. See "10.2.4 Viewing a Video File" on the facing page or"10.2.5 Receiving a Video Stream" on the facing page.

10.2.2 Streaming Video from a Drone

The drone can stream the full VR-Engage window or only the drone's camera view.

➤ **To stream video from a drone:**

1. Add or select an entity with a drone.
2. Engage the character.
3. Open the Action menu and choose **Deployable Objects > Deploy Spark**.
4. From the Drone submenu, choose **Streaming Video** and use an Action menu control to open the submenu. The Streaming Video submenu opens.
5. Use an Action menu control to toggle **Video Stream** to Started.

6. Choose **Video Source** and use an Action menu control to open the submenu. The Video Source menu opens.
7. Select whether the source of the video stream should be the Main Window or the Drone View.

   A user can then view the file or receive the video stream. See "10.2.4 Viewing a Video File" below or"10.2.5 Receiving a Video Stream" below.

10.2.3 Specifying the Video Stream File Settings

VR-Engage comes with a list of pre-existing configurations for streaming video. You can access the list of available configurations either by choosing **Settings > Video Streams** in Admin mode or by looking in *./appData/videoStreams*. The default stream configuration is called *H264 HW-MPGTS-FILE*, which instructs VR-Engage to stream H264-encoded video to a file in *./userData/videoCaptures*.

You can specify which configuration VR-Engage should use by editing the **DtStreamingVideoUpdater** component in *./data/simulationModelSets/VR-Engage/roles/sensorOperator.lua* or *./data/simulationModelSets/VR-Engage/roles/devices/deployableObjects.lua* and specifying a video stream configuration.

> If you want to modify any of the pre-existing video stream configurations or create a new one, see the chapter *Using Streaming Video* in the *VR-Vantage Users Guide*.

10.2.4 Viewing a Video File

When you save a video stream to a file, you can open it in a media player such as VLC media player. By default, you can find your captured streaming video files in *./userData/videoCaptures*. Files streamed from a sensor operator begin with "sensorOperator" and files streamed from a drone begin with "drone".

10.2.5 Receiving a Video Stream

If you have configured VR-Engage to stream video over a network, you must have a viewing application installed on the machine where you want to receive and view the video.

➤ **To receive the video stream:**

1. Launch the viewing application.
2. Configure the application with the same network parameters, such as the UDP port, as those specified in the video streaming configuration that you specified VR-Engage should use.

As long as the video stream is turned on for the sensor or drone, the video stream should display in the viewing application.

10.3. Capturing Images

The sensor operator and drone have cameras that can take screen shots of the camera view and save them to your computer. When you save a screen shot, it is saved in *./userData/screenshots* as a *PNG* file.

➤ **To take pictures with the sensor operator:**

1. Add or select an entity that has a sensor.
2. Engage as the Sensor Operator.
3. Open the Action menu and choose **Gimbaled Sensor > Capture Image**.

➤ **To take pictures with the drone:**

1. Add or select a character with a drone.
2. Engage the character.
3. Open the Action menu and choose **Deployable Objects > Deploy Spark**.
4. Fly the drone to the location where you want to take a picture.
5. Open the Action menu and choose **Drone > Capture Image**.



You can also capture images using the B button on the controller.



11. Mouse, Keypad, and Controller Mappings

This chapter describes mouse and keyboard mappings and controller mappings for VR-Engage. For details about configuration additional controllers, see "11.12. Configuring Games Controllers and Keyboard Control" on page 219.

11.1. Mouse and Keyboard Mappings 191
- 11.1.1 Application Management 191
- 11.1.2 Action Menu Controls 191
- 11.1.3 Human Controls 192
- 11.1.4 Ground Vehicle Controls 193
- 11.1.5 Aircraft Controls 194

11.2. Game Controller Mappings for Using the Action Menu 195
11.3. Game Controller Mappings for Human Control 196
11.4. Game Controller Mappings for Drones 197
11.5. Game Controller Mappings for the LAV and Tanks 197
11.6. Game Controller Mappings for Cars and Trucks 199
11.7. Racing Wheel Controllers 199
11.8. Flight Controllers 201
- 11.8.1 HOTAS Warthog Mappings for Fixed-Wing Entities 202
- 11.8.2 HOTAS Warthog Mappings for Rotary-Wing Entities 205
- 11.8.3 T.Flight Hotas X Mappings for Fixed-Wing Entities 207
- 11.8.4 T.Flight Hotas X Mappings for Rotary-Wing Entities 210
- 11.8.5 Game Controller Flight Control Mappings for Fixed-Wing Entities 213
- 11.8.6 Game Controller Flight Control Mappings for Rotary-Wing Entities 214
- 11.8.7 Game Controller Mappings for the Sensor Operator 214

11.9. Head-Mounted Display Controllers 215
11.10. Game Controller Button Numbering 217
11.11. Tips for Logitech Gamepad 219
11.12. Configuring Games Controllers and Keyboard Control 219
- 11.12.1 Windows Game Controllers 220

11.12.2 File Structure 220
11.12.3 Mappings 221
11.12.4 Input Types 223
11.12.5 Adding a New Input Device 225
11.12.6 Player Controls State 227
11.12.7 Component Mapping Groups 229
11.12.8 List of Actions 230

11.1. Mouse and Keyboard Mappings

Using the mouse and keyboard, you can control human characters, ground vehicles, and navigate the Action menu. You can also perform some aircraft behaviors.

11.1.1 Application Management

Table 5 lists mouse and keyboard mappings for all roles in VR-Engage.

Table 5. Keyboard controls in all roles

Function	Key
Open display of gamepad and keyboard controls	F1
Open HTML help	Shift+F1
Toggle Action menu	~ (tilde)
Toggle full screen	Ctrl+Enter
Disable/enable mouse control	Ctrl
Exit to Application Menu / close Action menu	Esc
Toggle between Admin and User modes	Shift+F12
Reset to starting position (Admin mode only, scenario continues to run)	F5
Reset in current position (Admin mode only, scenario continues to run)	Ctrl+F5

11.1.2 Action Menu Controls

Table 6 lists keyboard controls for the Action menu.



You can use the corresponding number to select items in the Action menu, but that this number changes when you scroll through the list. For example, 1 does not always toggle the weapon state.

Table 6. Action menu mouse and keyboard mappings

Mouse Mappings	
Select menu item	Number matching menu item Mouse click (when mouse control is disabled)
Keyboard Mappings	
Toggle Action menu	~ (tilde)
Next menu item	Up arrow
Previous menu item	Down arrow
Select menu item	Right arrow or Enter
Return to previous menu	Left arrow

11.1.3 Human Controls

Table 7 lists mouse and keyboard mappings for human players.

Table 7. Human player mouse and keyboard controls

Function	Key / mouse movement
Mouse Mappings	
Turn / move weapon	mouse move
Aim	right mouse button
Fire	left mouse button
Zoom in (binoculars / minimap)	mouse wheel forward
Zoom out (binoculars / minimap)	mouse wheel backward
Keyboard Mappings	
Movement	
Run forward	W
Sprint forward	left Shift+W
Walk forward	left Alt+W
Run backward	S
Sprint backward	left Shift+S
Walk backward	left Alt+S
Turn left	H
Turn right	K
Posture up	Q
Posture down	E
Stand	C
Lay prone	Z
Toggle view mode (first- / third-person)	V
Parachuting	
Deploy parachute	Spacebar
Steer left or right	A or D
Slow rate of descent	A+D
Engagement	
Aim	, (comma)
Strafe left	A
Strafe right	D
Pitch weapon/look up	U
Pitch weapon/look down	J

Table 7. Human player mouse and keyboard controls (continued)

Function	Key / mouse movement
Toggle Weapon State (Stowed, Ready, Aiming)	' (apostrophe)
Cycle weapons	Spacebar
Select weapon	1, 2, 3, and so on (not on number pad)
Reload weapon	R
View and Accessories	
Look (weapons are disabled while looking)	. (period) + mouse or H, K, U, and J keys
Toggle night vision goggles	N
Toggle binoculars	B
Zoom in	=
Zoom out	- (dash, not on number pad)
Talk on the radio	T
Toggle flashlight	F
Toggle compass	G
Toggle minimap	M
Zoom in on minimap	+ (plus sign on number pad)
Zoom out of minimap	- (minus sign on number pad)
Reset minimap zoom	0 (number pad)
Toggle HUD (weapon, health, and so on)	/

11.1.4 Ground Vehicle Controls

Table 8 lists keyboard controls for ground vehicles.

Table 8. Keyboard mappings for ground vehicles

Function	Key / mouse movement
Movement and view (driver role)	
Start/stop engine	X
Throttle	W
Brake	S
Steer left	A
Steer right	D
Shift gear up	Q
Shift gear down	E

Table 8. Keyboard mappings for ground vehicles (continued)

Function	Key / mouse movement
Parking brake	P
Headlights	L
Differential lock (LAV) Pivot (tanks)	C
Cycle saved views	V
Specify saved view	1, 2, 3, 4, 5, 6 (as defined)
Engagement (gunner and driver/gunner roles)	
Move turret (gunner)	mouse move
Fire weapon (gunner)	left mouse button
Cycle sensor	M
Turret left	H
Turret right	K
Turret up	U
Turret down	J
Next weapon	R
Previous weapon	V
Fire	F
Commander RWS role	
Look	mouse move or H, K, U, and J keys
Toggle laser designator	L
Slew to cue	C
Gunner override	O
Cycle sensor	M
Reset sensor	R
Zoom Mappings (commander and gunner only)	
Zoom in	I / mouse wheel forward
Zoom out	Y / mouse wheel backward

11.1.5 Aircraft Controls

Table 9 lists the keyboard mappings for aircraft. You require another controller to fly the aircraft. See "11.8. Flight Controllers" on page 201.

Table 9. Keyboard mappings for fixed-wing aircraft

Function	Key
Toggle radar warning receiver	M
Observer mode	V
Toggle landing gear	G
Suppress warnings	Spacebar
Engagement	
Cycle weapons	A
Move radar cursor up	I
Move radar cursor down	K
Move radar cursor left	J
Move radar cursor right	L
Select target closest to cursor	Z
Unselect target closest to cursor	X
Cycle target list backward	Y
Cycle target list forward	H
Clear air target list	U
Countermeasures	
Launch chaff	C
Launch flares	F

11.2. Game Controller Mappings for Using the Action Menu

Figure 74 on the next page illustrates the game controller numbering and naming conventions.

Table 10 lists the controls for using the Action menu.

Table 10. Game controller mappings - Action menu

Function	Game Controller Key / Button	Button / Axis
Toggle Action menu	Back button	Button 4
Move in menu	D-pad	Buttons 11, 12, 13, and 14
Select	A	Button 0

11.3. Game Controller Mappings for Human Control

Figure 74 illustrates the game controller numbering and naming conventions.

Figure 74. Generic game controller button conventions

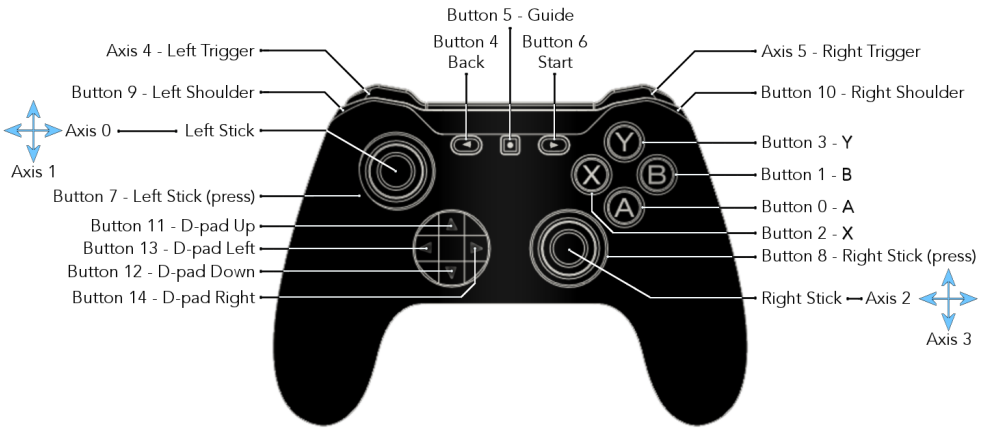


Table 11 lists the game controller buttons for human characters.

Table 11. Game controller mappings - human

Function	Game Controller Key / Button	Button / Axis
Movement and view		
Move	left thumbstick	Axis 0 and Axis 1
Sprint	left thumbstick (press)	Button 7
Rotate eyepoint	right thumbstick	Axis 2 and Axis 3
Free look	right thumbstick (press)	Button 8
Posture select	B button	Button 1
Toggle binoculars	X button	Button 2
Zoom binoculars in or out	right shoulder button left shoulder button	Buttons 9 and 10
Talk on radio	A button	Button 0
Toggle night vision goggles	D-pad down	Button 12
Human Combat		
Aim	left trigger	Axis 4
Fire/lase target	right trigger	Axis 5
Cycle weapons	Y button	Button 3
Parachuting (paratroopers only)		
Deploy parachute	A button	Button 0

Table 11. Game controller mappings - human (continued)

Function	Game Controller Key / Button	Button / Axis
Steer left or right	left or right shoulder button	Button 9 or 10
Slow rate of descent	both shoulder buttons simultaneously	Buttons 9 and 10

11.4. Game Controller Mappings for Drones

These controls are for the drone that can be deployed by human characters: Civilian Female With Drone, Civilian Male With Drone, and Police Officer With Drone. Figure 74 on the previous page illustrates the game controller numbering and naming conventions.

Table 12 lists the controls for the drone.

Table 12. Game controller mappings - drone

Function	Game Controller Key / Button	Button / Axis
Drone Flight Controls		
Altitude	left thumbstick (forward/back)	Axis 1
Direction Facing	left thumbstick (left/right)	Axis 0
Movement	right thumbstick	Axis 2 and Axis 3
Brake	A button	Button 0
Drone Camera Controls		
Zoom in	right shoulder button	Button 9
Zoom out	left shoulder button	Button 10
Capture Image	B button	Button 1
Pan up or down	D-pad (up/down)	Buttons 11 and 12

11.5. Game Controller Mappings for the LAV and Tanks

Figure 74 on the previous page illustrates the game controller numbering and naming conventions.

Table 13 lists the driver's controls for the LAV and tanks.

Table 13. Driver game controller mappings - LAV and tanks

Function	Game Controller Key / Button	Button / Axis
Vehicle Control		
Start/stop engine	Start button	Button 6
Throttle	left thumbstick	Axis 1

Table 13. Driver game controller mappings - LAV and tanks (continued)

Function	Game Controller Key / Button	Button / Axis
Shift gear up	right shoulder button	Button 10
Shift gear down	left shoulder button	Button 9
Steering	right thumbstick	Axis 2
Differential lock (LAV) Pivot (tanks)	X button	Button 2
Headlights	A button	Button 0
Parking brake	left thumbstick (press)	Button 7
Cycle saved views	Y button	Button 3
Engagement		
Change sensor	right thumbstick (press)	Button 8
Turret (gunner) (gunner/driver)	right thumbstick D-pad	Axis 2 and 3 Buttons 11, 12, 13, and 14
Cycle weapon (driver)	left trigger	Axis 4
Next weapon (gunner)	left trigger	Axis 4
Previous weapon (gunner)	D-pad up	Button 11
Fire weapon	right trigger	Axis 5

Table 14 lists the commander's controls for the LAV and tanks. (Not all tanks are configured with commander roles.)

Table 14. Commander game controller mappings - LAV and tanks

Function	Game Controller Key / Button	Button / Axis
Rotate/elevate camera	right thumbstick	Axis 2 and Axis 3
Change sensor	D-pad down	Button 14
Zoom out	left shoulder button	Button 9
Zoom in	right shoulder button	Button 10
Slew to cue (Commander RWS only)	X button	Button 2
Gunner override (Commander RWS only)	Y button	Button 3
Toggle Laser Designator (Commander RWS only)	A button	Button 0

Table 15 lists the gunner's controls for the LAV and tanks.

Table 15. Gunner game controller mappings - LAV and tanks

Function	Game Controller Key / Button	Button / Axis
Rotate/elevate gun	right thumbstick	Axis 2 and Axis 3
Change sensor	right thumbstick (press)	Button 8
Zoom out	left shoulder button	Button 9
Zoom in	right shoulder button	Button 10
Next weapon	left trigger	Axis 4
Previous weapon	D-pad up	Button 11
Fire weapon	right trigger	Axis 5

11.6. Game Controller Mappings for Cars and Trucks

This topic describes game controller mapping for cars and trucks. These vehicles have a driver role, but for the most part do not have weapons. Figure 74 on page 196 illustrates the game controller numbering and naming conventions.

Table 16 lists the driver’s controls for cars and trucks.

Table 16. Driver game controller mappings - cars and trucks

Function	Game Controller Key / Button	Button / Axis
Vehicle Control		
Start/stop engine	Start button	Button 6
Throttle	left thumbstick	Axis 1
Shift gear up	right shoulder button	Button 10
Shift gear down	left shoulder button	Button 9
Parking Brake	left thumbstick (press)	Button 7
Steering	right thumbstick	Axis 2
Headlights	A button	Button 0

11.7. Racing Wheel Controllers

VR-Engage supports the Logitech G27 and G29 racing wheel controllers.

If your racing wheel is not working as expected, make sure you are using the latest drivers.

Figure 75 illustrates the mappings of the G27 racing wheel.

Figure 75. G27 racing wheel controls



Table 17 lists the G27 racing wheel controls.

Table 17. G27 Racing Wheel control mappings

Function	Key/Button
Next View	Button 4 Down Toggle
Previous View	Button 3 Down Toggle
Cycle Sensor	Button 16 Down Toggle
Brake	Axis 3
Differential Lock	Button 7 Down Toggle
Engine Enable	Button 8 Down Toggle
Gear Down	Button 6 down Toggle
Gear Up	Button 5 Down Toggle
Headlights	Button 20 Down Toggle
Parking Brake	Button 22 Down Toggle
Reset	Button 23 Down Toggle
Steering	Axis 1
Throttle	Axis 2

Figure 76 illustrates the mappings of the G29 racing wheel.

Figure 76. G29 racing wheel controls

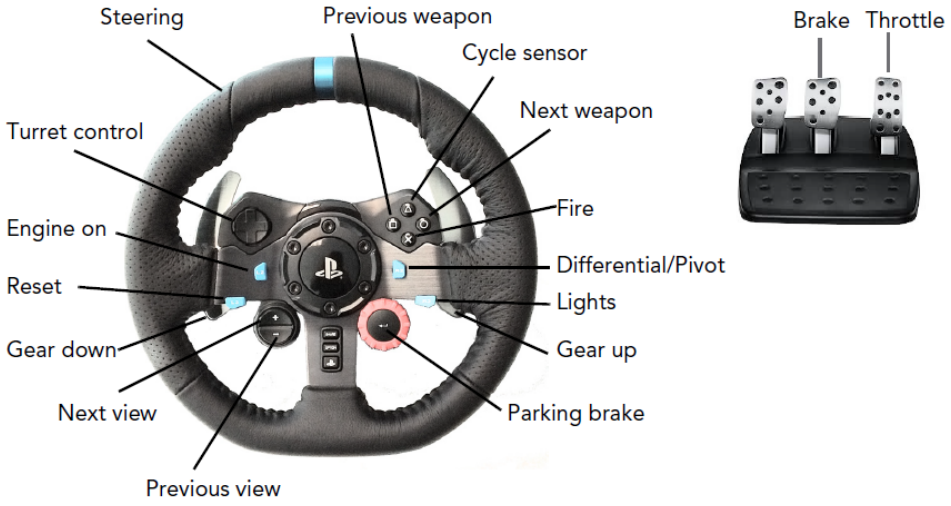


Table 18 lists the G29 racing wheel controls.

Table 18. G29 racing wheel control mappings

Function	Key/Button
Next View	Button 20 Down Toggle
Previous View	Button 21 Down Toggle
Cycle Sensor	Button 4 Down Toggle
Brake	Axis 3
Differential Lock	Button 7 Down Toggle
Engine Enable	Button 8 Down Toggle
Gear Down	Button 6 down Toggle
Gear Up	Button 5 Down Toggle
Headlights	Button 11 Down Toggle
Parking Brake	Button 24 Down Toggle
Reset	Button 12 Down Toggle
Steering	Axis 1
Throttle	Axis 2
Differential Lock	Button 7 Down Toggle

11.8. Flight Controllers

VR-Engage includes mappings for the Thrustmaster HOTAS Warthog flight (Joystick-HOTAS Warthog) controller, the T.Flight Hotas X flight controller, the Gamepad F310 and F710 (Logitech Dual Action), and Xbox controllers.

11.8.1 HOTAS Warthog Mappings for Fixed-Wing Entities

Figure 77 and Figure 78 show the mappings for the Thrustmaster Warthog flight controller for fixed-wing entities.

Figure 77. Thrustmaster HOTAS Warthog joystick mappings - fixed-wing



Figure 78. Thrustmaster HOTAS Warthog throttle mappings - fixed-wing

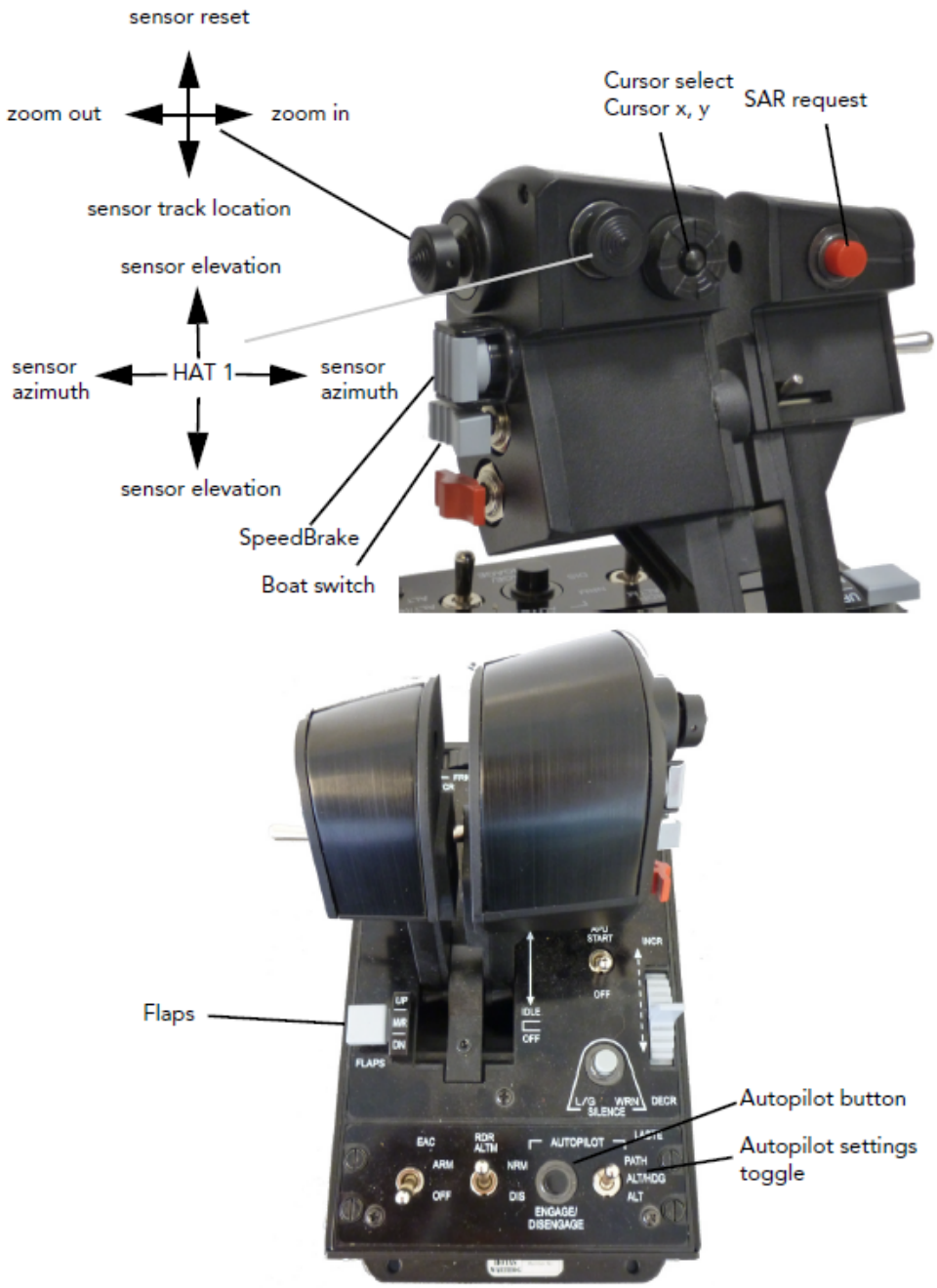


Table 19 lists the flight controls for the Thrustmaster HOTAS Warthog controller for fixed-wing entities.

Table 19. Thrustmaster HOTAS Warthog mappings - fixed-wing

Function	Key/Button
Extended Flight Controls	
AutopilotMode1	Throttle Button 27 Down Toggle
AutopilotMode2	Throttle Button 28 Down Toggle
AutopilotToggle	Throttle Button 26 Down Toggle
Flap 1	Throttle Button 22 Down Toggle. Toggle states match joystick up, maneuver, down.
Flap 2	Throttle Button 23 Down Toggle. Toggle states match joystick up, maneuver, down.
NoseWheelSteering	Button 3 Down Toggle
ToggleLandingGear	G Toggle
Engagement	
cycle weapon	Throttle Button 5 Down Toggle
fire selected weapon	Throttle Button 1 Down Toggle
replenish fuel and ordnance	Throttle Button 4 Down Toggle
FixedWingLib CGF Flight Controls	
air brake	Boat Switch (Button 10) Down. This causes excess drag and the plane's speed drops quickly.
pitch	Joystick forward and back (Axis 2)
roll	Joystick left and right (Axis 1)
throttle	Throttle lever (Axis 3)
wheel brake	SpeedBrake (Button 7) Down, button 8 Down. If you pull back and release, it re-centers. If you push forward, it locks in place. Both directions activate full brakes.
Gimbaled Sensor	
sensor-azimuth	Throttle HAT 1 Right (Up), HAT 1 Left (Down)
sensor-cycle	Joystick Button 7 Down Toggle
sensor-elevation	Throttle HAT 1 Up (Up), HAT 1 Down (Down)
sensor-reset	Throttle Button 3 Down
sensor-track-location	Throttle Button 5 Down
sensor-zoom	Throttle Button 4 (Down), Button 6 (Up)
Observer Control	
ObserverAttachMode	V (on the keyboard)
Radar Scope	
cursor-select	Button 1 (also clears the selection from the air target list)

Table 19. Thrustmaster HOTAS Warthog mappings - fixed-wing (continued)

Function	Key/Button
cursor-x	Axis 1
cursor-y	Axis 2
Clear air target list	U (on the keyboard)
SAR	
request	Throttle Button 15

11.8.2 HOTAS Warthog Mappings for Rotary-Wing Entities

Figure 79 and Figure 80 show the mappings for the Thrustmaster Warthog flight controller for rotary-wing entities.

Figure 79. Thrustmaster HOTAS Warthog joystick mappings - rotary-wing



Figure 80. Thrustmaster HOTAS Warthog throttle mappings - rotary-wing

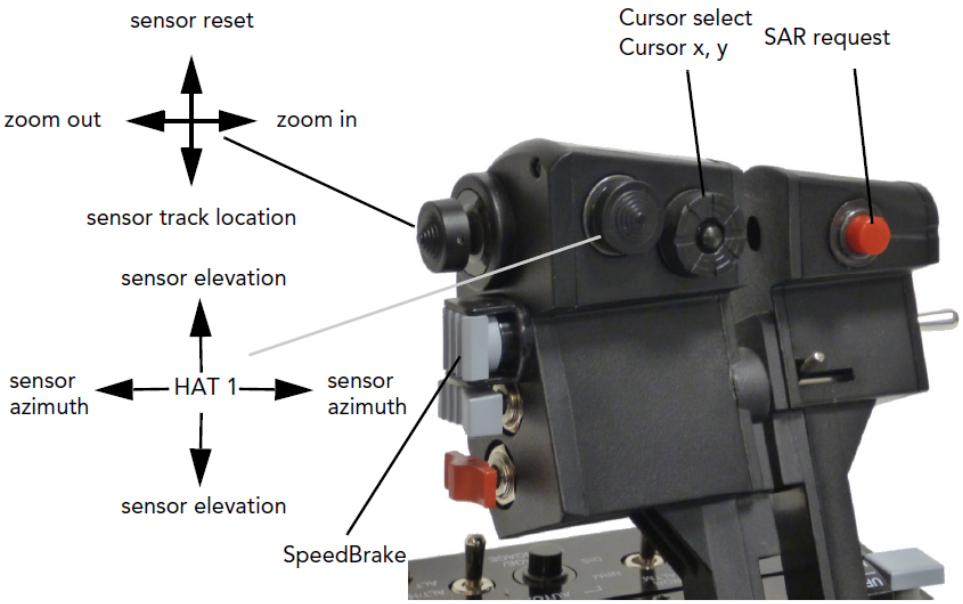


Table 20 lists the flight controls for the Thrustmaster HOTAS Warthog controller for rotary-wing entities.

Table 20. Thrustmaster HOTAS Warthog mappings - rotary-wing

Function	Key/Button
Engagement	
fire selected weapon	Button 1 Down Toggle
cycle weapon	Button 5 Down Toggle
replenish fuel and ordnance	Button 4 Down Toggle
Gimbaled Sensor	
sensor-azimuth	Throttle HAT 1 Right (Up), HAT 1 Left (Down)
sensor-cycle	Joystick Button 7 Down Toggle
sensor-elevation	Throttle HAT 1 Up (Up), HAT 1 Down (Down)
sensor-reset	Throttle Button 3 Down

Table 20. Thrustmaster HOTAS Warthog mappings - rotary-wing (continued)

Function	Key/Button
sensor-track-location	Throttle Button 5 Down
sensor-zoom	Throttle Button 4 (Down), Button 6 (Up)
Radar Scope	
cursor-select	Button 1 (also removes selection from the air target list)
cursor-x	Axis 1
cursor-y	Axis 2
Clear air target list	U (on keyboard)
RotorLib CGF Rotary Wing Controls	
collective	Axis 3, <UP>, <DOWN>
pedal	HAT 1 Up, HAT 1 Left
pitch	Axis 2, W
roll	Axis 1, D (Up), A (Down)
wheel brake	SpeedBrake (Button 7) Down, button 8 Down. If you pull back and release, it recenters. If you push forward, it locks in place. Both directions activate full brakes.
SAR	
request	Button 15

11.8.3 T.Flight Hotas X Mappings for Fixed-Wing Entities

Figure 81 shows the joystick controls for the T.Flight Hotas X flight controller for fixed-wing entities.

VR-Engage also supports the T.Flight Hotas 4 controller. The button layout is similar to the T.Flight Hotas X.

Figure 81. T.Flight Hotas X joystick - fixed wing



Figure 82 shows the throttle controls for the T.Flight Hotas X flight controller for fixed-wing entities.

Figure 82. T.Flight Hotas X throttle - fixed-wing

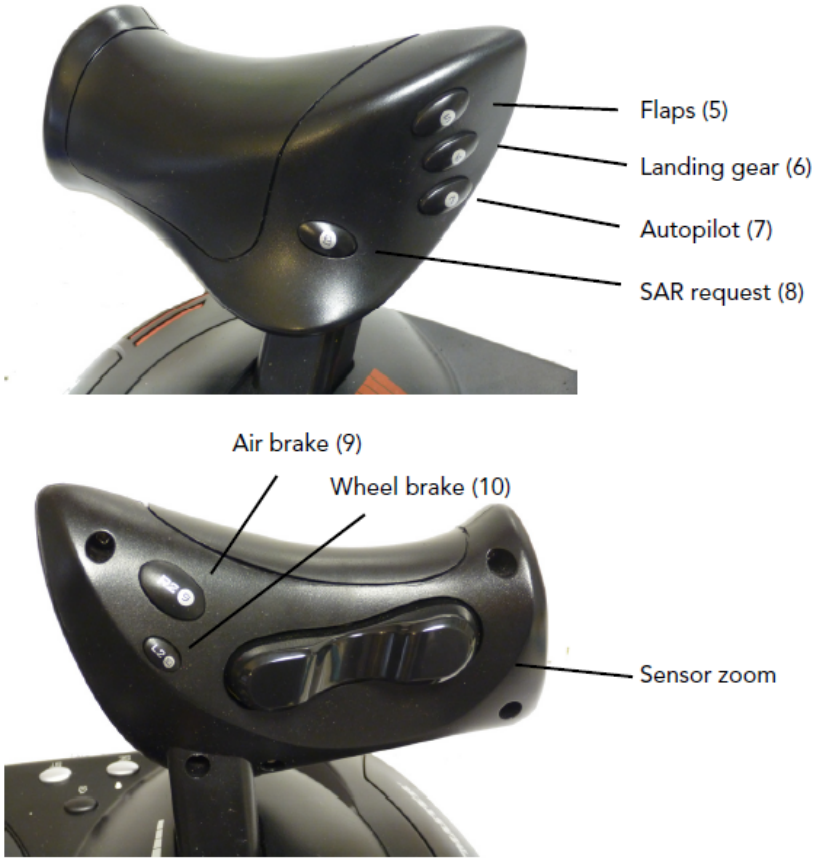


Table 21 lists the flight controls for the T.Flight Hotas X controller for fixed-wing entities.

Table 21. T.Flight Hotas X mappings - fixed-wing

Function	Key/Button
Engagement	
cycle weapon	Button 5 Down Toggle
fire selected weapon	Button 1 Down Toggle
cursor select target	Button 3 Down Toggle
FixedWingLib CGF Flight Controls	
pedal	Joystick twist left and right (Axis 4)
pitch	Joystick forward and back (Axis 2)
roll	Joystick left and right (Axis 1)
throttle	Throttle forward and back (Axis 3)
autopilot	Button 7
air brake	Button 9 Down

Table 21. T.Flight Hotas X mappings - fixed-wing (continued)

Function	Key/Button
landing gear	Button 6
flaps	Button 5
wheel brake	Button 10 Down
Menu	
Action menu	Button 4 Down
down	HAT 1 Down, <DOWN>
left	HAT 1 Left, <LEFT>
right	HAT 1 Right, <RIGHT>
up	HAT 1 Up, <UP>
Observer Control	
Attach mode	V (on keyboard)
Radar Scope	
down	HAT 1 Down, <DOWN>
left	HAT 1 Left, <LEFT>
right	HAT 1 Right, <RIGHT>
up	HAT 1 Up, <UP>
select target closest to cursor	Z (on keyboard)
clear current radar selection	X (on keyboard)
Release Bomb	
release	Button 1 Down
SAR	
request	S, Button 8 Down

11.8.4 T.Flight Hotas X Mappings for Rotary-Wing Entities

Figure 83 shows the joystick controls for the T.Flight Hotas X flight controller for rotary-wing entities.

VR-Engage also supports the T.Flight Hotas 4 controller. The button layout is similar to the T.Flight Hotas X.

Figure 83. T.Flight Hotas X joystick - rotary-wing



Figure 84 shows the throttle controls for the T.Flight Hotas X flight controller for rotary-wing entities.

Figure 84. T.Flight Hotas X throttle - rotary-wing



Table 22 lists the flight controls for the T.Flight Hotas X controller for rotary-wing entities.

Table 22. T.Flight Hotas X mappings - rotary-wing

Function	Key/Button
Engagement	
cycle weapon	Button 5 Down Toggle
fire selected weapon	Button 1 Down Toggle
cursor select target	Button 3 Down Toggle
Menu	
Action menu	Button 4 Down
down	HAT 1 Down, <DOWN>
left	HAT 1 Left, <LEFT>
right	HAT 1 Right, <RIGHT>
up	HAT 1 Up, <UP>

Table 22. T.Flight Hotas X mappings - rotary-wing (continued)

Function	Key/Button
Radar Scope	
down	HAT 1 Down, <DOWN>
left	HAT 1 Left, <LEFT>
right	HAT 1 Right, <RIGHT>
up	HAT 1 Up, <UP>
select target closest to cursor	Z (on keyboard)
clear current radar selection	X (on keyboard)
RotorLib CGF Rotary Wing Controls	
collective	Throttle forward and back (Axis 3)
pedal	Joystick twist left and right (Axis 4)
pitch	Joystick forward and back (Axis 2)
roll	Joystick left and right (Axis 1)
wheel brake	Button 10 Down
SAR	
request	S, Button 8

11.8.5 Game Controller Flight Control Mappings for Fixed-Wing Entities

Figure 74 on page 196 illustrates the game controller numbering and naming conventions.

Table 23 lists the flight controls for fixed-wing entities.

Table 23. Game controller flight control mappings - fixed-wing

Function	Game Controller Key / Button	Button / Axis
Engagement		
Cycle weapons	left trigger	Axis 4
Fire weapon	right trigger	Axis 5
Radar cursor select	left shoulder button	Button 9
Radar cursor	D-pad	Buttons 11, 12, 13, 14
Flight Controls		
Pedal	left thumbstick (forward/back)	Axis 0
Throttle	left thumbstick (forward/back)	Axis 1

Table 23. Game controller flight control mappings - fixed-wing (continued)

Function	Game Controller Key / Button	Button / Axis
Pitch	right thumbstick (forward/back)	Axis 3
Roll	right thumbstick (left/right)	Axis 2
Air brakes	A button	Button 11
Wheel brakes	X button	Button 13
Flaps	Y button	Button 0
Auto pilot	right shoulder button	Button 10

11.8.6 Game Controller Flight Control Mappings for Rotary-Wing Entities

Figure 74 on page 196 illustrates the game controller numbering and naming conventions.

Table 24 lists the flight controls for rotary-wing entities

Table 24. Game controller flight control mappings - rotary-wing

Function	Game Controller Key / Button	Button / Axis
Rotary Wing Controls		
Collective	Left Stick (left/right)	Axis 0
Pedal	Right Stick (forward/back)	Axis 1
Pitch	Right Stick (left/right)	Axis 2
Roll	Left Stick (forward/back)	Axis 3
Engagement		
Cycle Weapon	Left Trigger	Axis 4
Fire Weapon	Right Trigger	Axis 5

11.8.7 Game Controller Mappings for the Sensor Operator

Figure 85 shows the buttons for the F310 Gamepad for the sensor operator role.

Figure 85. Gamepad sensor operator controls



Table 25 lists the controls for the sensor operator

Table 25. Game controller mappings - sensor operator

Function	Gamepad Key/Button	Xbox Controller Key/Button
Sensor orientation	Axis 1 and Axis 2	Axis 1 and Axis 2
Sensor mode	Button 1	Button 13
Track location	Button 2	Button 11
Track the entity closest to the center of the screen	Button 3	Button 12
Reset sensor	Button 4	Button 14
Zoom	Axis 4	Axis 4

11.9. Head-Mounted Display Controllers

The Oculus Rift and HTC Vive head-mounted displays support hand controllers. This section shows the mappings for these controllers, which can be used for soldier roles. If you do not have hand controllers, you can use a keyboard, Xbox controller, or Logitech Gamepad. (HMD hand controllers are not mapped in joystick settings.)

Figure 86 illustrates the controls on the Oculus Rift touch controller.

Figure 86. Oculus Rift touch controller

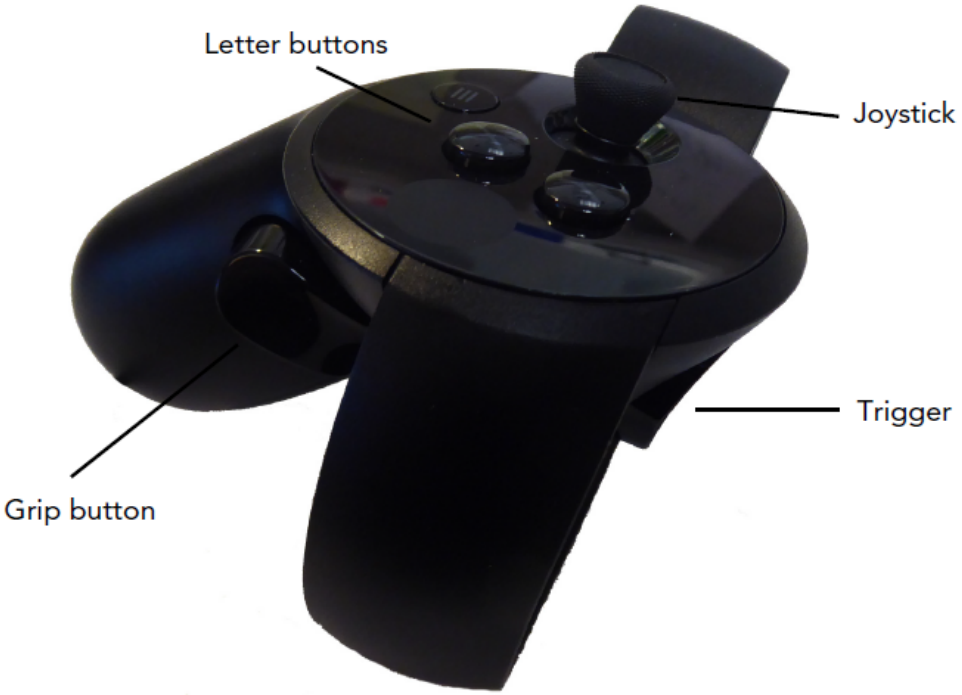
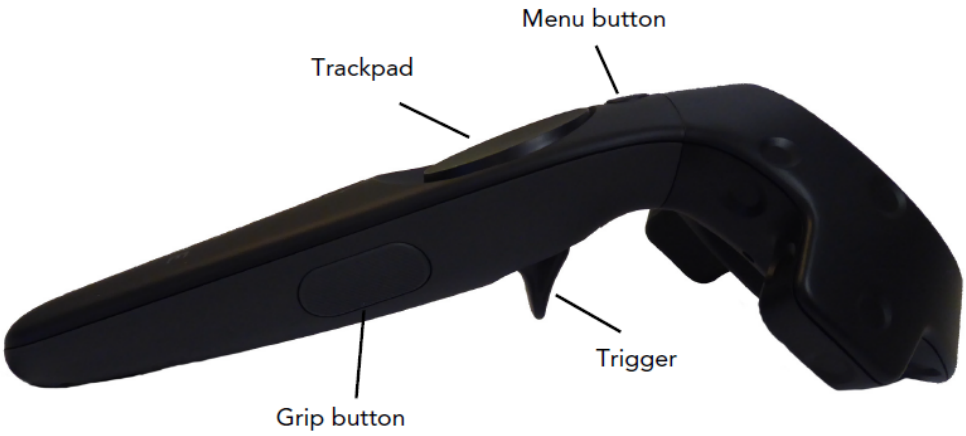


Figure 87 illustrates the controls on the Oculus Rift touch controller.

Figure 87. HTC Vive controller



The controls work as follows:

- Fire. Press the trigger on the dominant hand to fire the weapon.
- Aim. Press the grip button to raise the weapon to the aim-down-sights position.
- Move. Movement is controlled by the right hand joystick (Oculus Rift) or trackpad (HTC Vive), as follows:

  - Walk forward. Push/touch forward. Movement is based on the orientation of the entity, not the direction it is looking or aiming.
  - Walk backwards. Pull/touch backward.
  - Walk sideways. Push/touch left and right
  - Sprint. Push down while pressing forward.
- Turn. Press/touch the left hand joystick or trackpad to the left or right.
- Change posture. Press the trigger on the non-dominant hand for Posture Up. Press the grip button on the non-dominant hand for Posture Down.
- Action menu. To open, press Y on the non-dominant hand Oculus touch controller. Press the menu button on the non-dominant hand HTC Vive controller. While the menu is open, the joystick or trackpad acts as a D-pad.

11.10. Game Controller Button Numbering

Each trigger and joystick direction is considered to be an axis and can be mapped to a joystick control. Everything else is a button and can be mapped to a button control. Game controllers are mapped in *.lua* files. (For details, see "11.12. Configuring Games Controllers and Keyboard Control" on page 219.) The mappings are identified by button and axis numbers. Figure 88 and Figure 89 identify the button and axis numbering scheme for the Xbox and Gamepad controllers.

Figure 88. Xbox controller button and axis numbering



Figure 89. Logitech Gamepad button and axis numbering

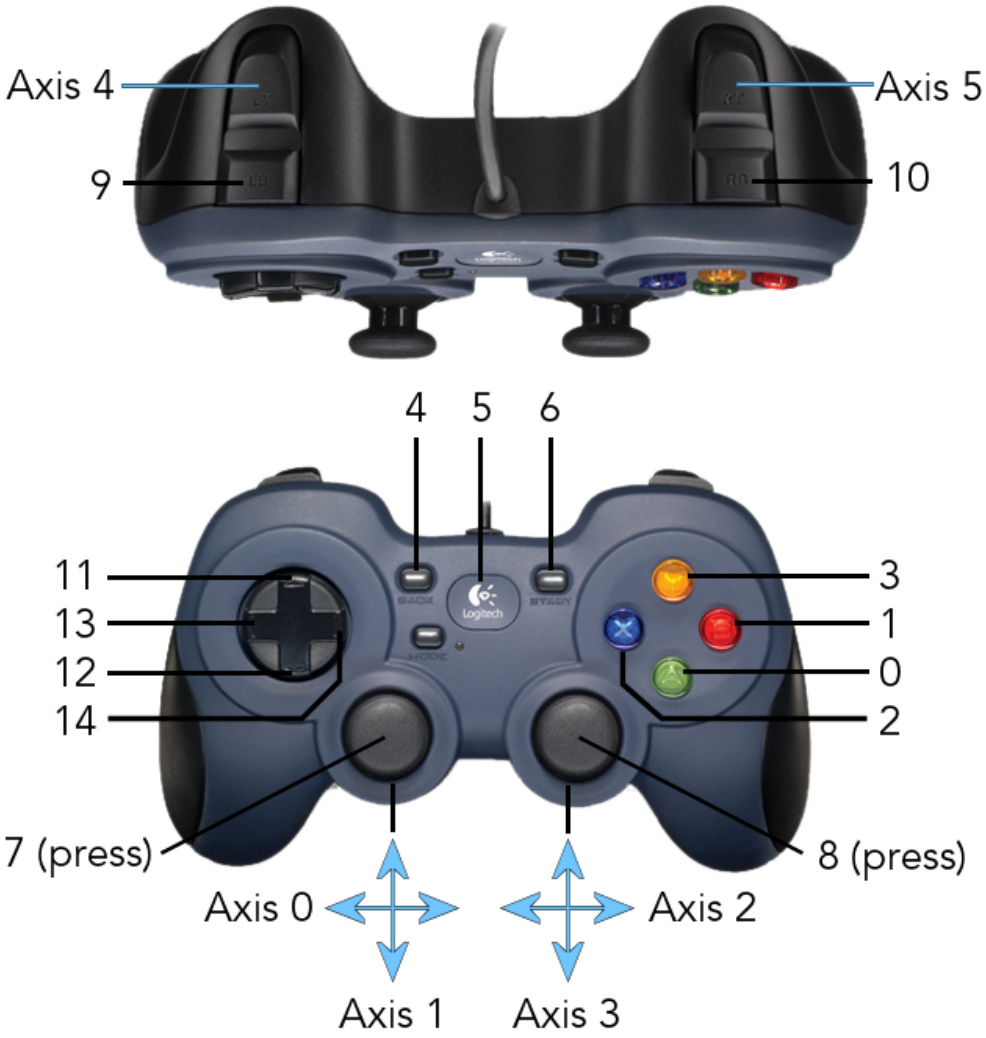


Table 26. Controller numbering

Number	Description
Button	
0	Xbox A or corresponding
1	Xbox B or corresponding
2	Xbox X or corresponding
3	Xbox Y or corresponding
4	Xbox Back or corresponding
5	Xbox Guide or corresponding
6	Xbox Start or corresponding
7	Left stick press
8	Right stick press

Table 26. Controller numbering (continued)

Number	Description
9	Left shoulder button
10	Right shoulder button
11	D-pad up
12	D-pad down
13	D-pad left
14	D-pad right
15	Miscellaneous (Xbox Series X share button, PlayStation 5 microphone button, Nintendo Switch Pro capture button, Amazon Luna microphone button)
16	Xbox Elite paddle P1
17	Xbox Elite paddle P1
18	Xbox Elite paddle P1
19	Xbox Elite paddle P1
20	PlayStation 4 / PlayStation 5 touchpad button
Axis	
0	Left stick x axis
1	Left stick Y axis
2	Right stick x axis
3	Right stick y axis
4	Left trigger
5	Right trigger

11.11. Tips for Logitech Gamepad

The Mode button switches the functions of the D-pad and the left thumbstick. Since it is easy to inadvertently press it, if the D-pad and thumbstick are not working as expected, check the Mode button. It should be off (no green light).

There is a switch on the back of the Gamepad that you can set to either D (DirectInput) or X (XInput). Either input should work to control VR-Engage entities as described in the previous sections. If you experience issues with X, set the controller to D.

11.12. Configuring Games Controllers and Keyboard Control

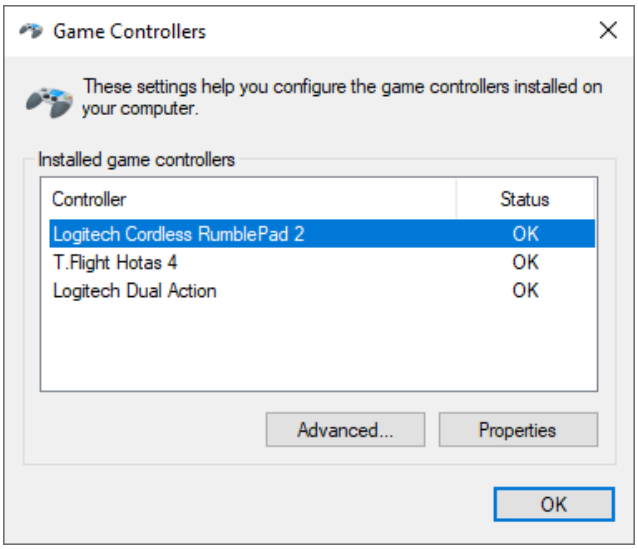
You can control entity movement and weapon fire using a game controller (such as a gamepad or joystick), a keyboard and mouse, or a combination of all of these. You can attach multiple joysticks and can use a mix of joystick and keyboard control at the same time. You configure joystick and keyboard mappings in *.lua* files.

11.12.1 Windows Game Controllers

VR-Engage requires that devices be a USB Windows Game Controller (as opposed to a USB human interface device) You can verify that a USB device is a game controller with the Game Controllers dialog. If the device is a game controller, you will see an entry for the device in that dialog when the device is connected (Figure 90).

➤ To open the Game Controllers dialog, in the Windows 10 search bar, enter Set Up USB Game Controllers and then select the entry.

Figure 90. Game Controllers dialog box



11.12.2 File Structure

Role Input Configuration Files

By default, all input configuration files are stored in *./data/simulationModelSets/VR-Engage/input*. You can change the location for these by specifying an alternate path in *./appData/scripts/playerStation.lua*:

```
  -- The path to the input directory for configuring VRE entities input
  inputSettingsPath = "$(DATA_DIR)/simulationModelSets/VR-
Engage/input/";
```

The configuration files at the top level typically represent the component input files, pointed to from the "inputConfigFile" field of the control logic components of the role configuration files. Some files closely align to roles, such as *driverInput.lua* and *gunnerInput.lua*. The *commonInput.lua* file maps functionality common to multiple roles, for example, navigating the Action menu, saved views, and mouse control.

Every component configuration file contains a single "configFiles" field with a list of input-device specific *.lua* configuration files.

This example is from *humanInput.lua*:

```
configFiles = {
  "human/gamepad.lua";
  "human/keyboard.lua";
  "human/mouse.lua";
}
```

Device Configuration Files

A device configuration file contains a single "inputDevices" field with a set of device configurations. The device configuration files are:

- DeviceName. Values are mouse, keyboard, joystick, or gamepad.
- DeviceType. This applies to joysticks and game controllers. Set this to configure mappings for a very specific device instead of all devices of that type, and should contain the device name, for example, "Logitech Extreme 3D".

  If there are both generic and specific mappings for a kind of input, the specific mapping is used. Generic device types are "gamepad" and "joystick". To create a custom mapping for a specific type of device, you must create an entry for it and set the DeviceType to the name of the device as it appears in the Game Controllers dialog box (Figure 90 on the previous page).
- MappingGroups. A set of mappings for the device's input (keyboard or controller) to some particular action in the software. The action could be an entity behavior, such as moving or shooting. It could also be an administrative action, such as releasing mouse control or opening and closing the Action menu.
  - Group. The mapping group name represents the context in which the mapped actions are used. For example, all actions in the "action-menu" group are only available when the user opens the Action menu.
  - Mappings. List of input-action mappings for the group.

11.12.3 Mappings

Common fields:

- Type. Describes the type of input event. Each device type supports different types of inputs:
  - Keyboard. key
  - Mouse. motion, button, double-click, scroll
  - Gamepad: axis, button (see "11.10. Game Controller Button Numbering" on page 217)
  - Joystick. axis, button, hat

  Each input type requires additional fields describing the specific input. For details on the input types, see "11.12.4 Input Types" on page 223.
- Action. The output or outcome of the input being triggered. See "11.12.8 List of Actions" on page 230.

These optional option fields manipulate the action frequency or the values:

- repeatRate (positive number). Intended to use with discrete inputs (buttons or keyboard keys). It emulates the input being repeatedly pressed and released according to the given rate in milliseconds for as long as it is being held.
- smoothing (Boolean). Used to simulate joystick movement with the mouse, for example, to make turret movement more realistic.
- valueTransforms. A list of functions for manipulating the output value. Each value transform has a type field, with parameter fields according to the type.

  When x denotes the raw input value:

Type	Relevant input type	Parameter fields	Output
dead-zone	axis	value (near 0)	$0\ if\ \lvert x\rvert \leq value,\ else\ x$
gain	axis	value (between 0 and 1)	$value \times x^3 + \left(1 - value\right) \times x$
invert	all	N/A	$-x$
scale	axis	value	$value \times x$
remap	axis	inputMin, inputMax, outputMin, outputMax	$outputMin + \frac{x - inputMin}{\lvert inputMax - inputMin\rvert} \times outputMax$

You can use multiple transforms at one time, even of the same type. The transforms are processed according to their order in the configuration file. For example, with value transforms of gain and scale in this order, the output is:

$$scale \times \left(gain \times x^3 + \left(1 - gain\right) \times x\right.$$

If the order is reversed, the output is:

$$gain \times \left(scale \times x\right)^3 + \left(1 - gain\right) \times scale \times x$$

For example:

```
{type="motion"; axis="y"; action="turret-elevation";
  valueTransforms = {
    {type="scale"; value=0.8};
  };
  smoothing=true;
};
```

11.12.4 Input Types

Button

Represents a pressable button on the mouse, gamepad, or joystick. For example:

```
{type="button"; id=0; action="fire";}
```

It requires the ID of the button on the device:

- Mouse: 0 for the left button and 1 for the right button. No other mouse buttons are supported.
- Gamepad: See "11.10. Game Controller Button Numbering" on page 217. If your gamepad has additional buttons, they cannot be mapped.
- Joystick. Button IDs are device specific.

Axis

Represents stick and trigger axis movements on the gamepad or joystick. For example:

```
{type="axis"; id=0; action="steering";}
```

It requires the ID of the axis on the device:

- Gamepad: See "11.10. Game Controller Button Numbering" on page 217. Only the axis shown in the figures and listed in the table are supported.
- Joystick: Axis IDs are device specific.

Moving a stick to the left or down is taken as negative movement of the axis, while right or up is positive movement. Triggers allow only positive movement.

Motion

Represents mouse movements, mostly used for mouse-look. Mouse motion is only processed when the user has mouse control enabled.

For example:

```
{type="motion"; axis="x"; action="turret-azimuth";}
```

Requires an axis field with a value of x or y.

Mouse Control

Mouse control is disabled by default. You can only enable mouse control when the `mouseControlSupport` configuration field of the control logic component in the role configuration file is "true". The control logic components are DtDriverControlLogic, DtHumanControlLogic, DtGunnerControlLogic, and so on.

When engaging with mouse control support, it can be toggled with the `toggle-mouse-control` action (**Alt**+**Enter** or **Ctrl**+left click) or enabled with enable-mouse control (double clicking).

Mouse control is automatically disabled when the VR-Engage window loses focus and is re-enabled when focus is restored, assuming it was enabled before losing focus.

Scroll

Represents mouse wheel scrolling. VR-Engage only supports the vertical axis of the first mouse wheel. If a mouse has multiple wheels or allows side-scrolling, those functionalities cannot be mapped.

For example:

```
{type="scroll"; action="sensor-zoom";}
```

Double-click

Requires the mouse button ID: `0` for the left button and `1` for the right button.



If "button" and "double-click" mappings share the same ID, double-clicking momentarily executes the "button" mapped action as well.

Key

Represents a key press on the keyboard. Requires a key field.

For example:

```
{type="key"; key="<F12>"; modifiers={"<SHIFT>"}; action="toggle-
application-mode"}
```

The key values that you can use are:

- Character keys. The character with letters in uppercase, for example, `1` or `A`.
- Keypad number. `<KP0>` through `<KP9>`.
- Functional keys. The keyname surrounded by angle brackets:
  - Arrow keys. `<UP>`, `<DOWN>`, `<LEFT>`, `<RIGHT>`
  - Function keys. `<F1>` through `<F12>`
  - Non-numeric number pad keys. `<SUBSTRACT>`, `<MULTIPLY>`, `<DIVIDE>`, `<DECIMAL>`
  - Modifier keys. `<LEFT SHIFT>`, `<RIGHT SHIFT>`, `<LEFT ALT>`, `<RIGHT ALT>`, `<LEFT CTRL>`, `<RIGHT CTRL>`
  - Others. `<BACKSPACE>`, `<TAB>`, `<CLEAR>`, `<PAUSE>`, `<ESC>`, `<SPACE>`, `<PAGE UP>`, `<PAGE DOWN>`, `<END>`, `<HOME>`, `<PRINT SCREEN>`, `<INSERT>`, `<DELETE>`

  Additional keys are not supported.

Modifiers is an optional field that allows you to map a conjunctional key and a modifier press to an action. The supported modifiers values are `<SHIFT>`, `<ALT>`, and `<CTRL>`. There is no distinction between left and right modifier presses. A mapping can have more than one modifier, for example:

```
key="A"; modifiers={"<CTRL>, <SHIFT>"};
```

maps **Ctrl**+**Shift**+**A**.

Hat

Represents use of a joystick. The ID field is optional and defaults to 0. The position value is required and represents the direction to which the hat switch is moved: up, right-up, right, right-down, down, left-down, left, left-up (when returned to the natural position).

For example:

```
{type="hat"; id=0; position="down"; action="next-item"}
```

11.12.5 Adding a New Input Device

The workflow to create plug-in for a new device is:

1. Create a class derived from `DtInputDevice`.
2. To load the input device, register it at `initPlayerStationModule` via `app->inputDeviceFactory()->addCreator`.
3. To send input data to the input manager, create a `DtInputData` object and process it with `myManager->processInput`.



There is no dependency between the device and input type. You can use any input type for the new device.

Gamepads

A gamepad is a device similar to an Xbox game controller. It includes a D-pad, two analog sticks, four buttons on the right-hand side, shoulder buttons, and triggers.

Most gamepads have built-in mappings. For example, the 0 axis represents the left stick's X axis and A represents the bottom button of the four buttons on the right on all Xbox controllers, PlayStation controllers, and Logitech F310 Gamepads.

Most gamepad devices should be automatically recognized as such and have built-in input mappings, therefore they should work with VR-Engage without requiring any additional configuration.

The gamepad mappings for most devices are defined in *./data/simulationModelSets/VR-Engage/inputgamecontrollerdb.txt*. If a device is recognized as a joystick instead of a gamepad, you must add an entry for the device to the mapping file. The file is community-maintained at https://github.com/gabomdq/SDL_GameControllerDB, which is frequently updated with new devices.



When you enable verbose printing to the console, there should be console output (print statements) for all connected gamepads and joysticks. The joystick printout includes its GUID, which you need to add it to the mapping file.

Creating a Unique Gamepad Mapping

You can map a specific gamepad differently than others. For example, you could have the axis and buttons activate different actions on the Logitech F310.

➤ **To create a unique gamepad mapping:**

1. Open the *gamepad.lua* file for the relevant component, for example, *./data/simulationModelSets/VR-Engage/input/human/gamepad.lua*.
2. Clone the existing generic gamepad input device entry.
3. Add a DeviceName="<*name*>" field, where <*name*> is the name of your device. You can find the device name from the verbose console output (print statements) when you add the controller, or by searching for the GUID in *./data/simulationModelSets/VR-Engage/inputgamecontrollerdb.txt*.
4. Modify the mappings.

The resulting file should look similar to:

```
inputDevices ={
  {
    DeviceType="gamepad";
    MappingGroups={
    ...
    };
  };
  {
    DeviceType="gamepad";
    DeviceName="Logitech F310";
    MappingGroups={
    ...
    };
  }
}
```

Alternatively, you can put the new inputDevice object in a different *.lua* file and link to it from the component *.lua* file.



Specific gamepad mappings completely override generic mappings. Ensure that you provide mappings for all actions you intend to support on a specific device.

Joysticks

Unlike gamepads, the IDs for joystick inputs can refer to different things across different devices. Each joystick device can have a different entry.

➤ **To add a joystick mapping:**

1. Open the *joystick.lua* file for the relevant component. If one does not exist, or you would like to use a different file, create or open the file and make sure that it is linked from the component lua file, for example, *./data/simulationModelSets/VR-Engage/input/driverGunnerInput.lua*.

2. Create an `inputDevices` object where `DeviceType="joystick"` and `DeviceName` is the driver device name, for example, "Logitech G HUB G29 Driving Force Racing Wheel USB". You can find the appropriate name in the **Windows Control Panel > Hardware and Sound > Devices and Printers**.
3. Add the relevant MappingGroups and mappings.

The resulting file should look similar to:

```
inputDevices ={
  {
    DeviceType="joystick";
    DeviceName="Logitech G HUB G29 Driving Force Racing Wheel USB"
    MappingGroups={
      {
        Group="driving";
        Mappings={
        ...
        };
      };
      {
        Group="engagement";
        Mappings={
        ...
        };
      };
    };
  };
}
```

11.12.6 Player Controls State

Player controls are enabled when:

- No modal is open.
- The scenario is playing.
- The entity is engaged.

When player controls are disabled, only loaded mapping layers that are always enabled remain active. Currently this applies to `app` and `entity-control` groups.

Mapping Conflicts

If the same input is mapped to different actions within the same group, the last mapping defined in the file is used.

Mapping groups are loaded in layers, with each layer (the last one loaded in most cases) having priority over the ones on the bottom. If the same input is mapped to different actions between different groups, the one on the top layer counts, assuming the action is applicable (that is, it has an action handler when the input is made). For example, when opening the Action menu while engaged as a human, pressing "down"

selects the next menu option instead of causing the human character to look down. This should be regarded when setting the priority of `ControlLogic` and `SimLogic` components in the role configuration files.

You can map multiple inputs to the same action.

Player controls are enabled when no modal is open, the scenario is playing, and the entity is engaged. (See `DtEngagedState::updatePlayerControlsState`).

When player controls are disabled, only loaded mapping layers that are always enabled remain active. Currently, this applies to the `app` and `entity-control` groups.

When players controls are disabled, mouse control is also turned off.

Raw Input Handling

The action input menu and SAR image request require raw input. When active (logic-wise—a raw input handler is set), they have priority over all existing mappings.

The implication is that, for example, if the user pressed A on the gamepad, the input menu processes the input and any mapping of A to an action is ignored.

Current Limitations

- Simultaneous activation of multiple inputs cannot be mapped to a single action, except when using keyboard modifiers.
- There is no distinction between left and right modifiers.
- Keyboard modifiers cannot be used with non-keyboard inputs. This means that you cannot map **Alt**+left click to a single action.
- If multiple joysticks or game controllers of the same type are connected, they must share the same mapping. You cannot have "X" activating "fire weapon" on the first connected Xbox controller and "reload" on the other.
- Non-standard keyboard keys, additional mouse buttons over the standard three, additional mouse wheels, and vertical scroll cannot be mapped.
- Controller touchpads and sensors are not supported.
- When the window is out of focus (even minimized/hidden), gamepad and joystick inputs continue to work. This does not apply to the mouse and keyboard.
- When mouse control is enabled and the Action menu is open, a user can select Action menu options with the mouse by moving the mouse fast enough. It is a bit difficult, depending on the window size, but can potentially happen by accident.

11.12.7 Component Mapping Groups

Table 27. Component mapping groups

Component name	Default config files	Group name	Loaded when
App/DtPlayerStationApp	appInput.lua (hardcoded)	App	Always
Common/DtMenu	commonInput.lua (hardcoded)	action-menu	Action/debug menu open
Common/DtEntityControlLogic		entity-control	Engaged
Common/DtSavedViewUpdater		view	Engaged
DtHumanInputLogic	humanInput.lua	general	Engaged
		combat	Engaged
		binoculars	Binoculars are used
		parachuting	Parachuting
		drone	Using deployed drone
DtCommanderControlLogic	commanderInput.lua lldrInput.lua	general	Engaged
DtDriverControlLogic	driverInput.lua driverGunnerInput.lua	driving	Engaged
DtGunnerControlLogic	gunnerInput.lua driverGunnerInput.lua	engagement	Engaged
		commander-rws	Using commander override (Commander RWS role)
DtLaserTargetDesignatorSimLogic	lldrInput.lua	laser-target-designator	Using LLDR
DtPilotSimLogic	pilotInput.lua	fixed-wing	Engaged fixed-wing
		rotary-wing	Engaged rotary-wing
		extended-flight-controls	Engaged
		instruments	Engaged

Table 27. Component mapping groups (continued)

Component name	Default config files	Group name	Loaded when
DtSensorOperatorInputLogic	sensorOperatorInput.lua	general	Engaged
DtSpectatorSimLogic	spectatorInput.lua	general	Engaged

11.12.8 List of Actions

Binary actions only take values of 0 and 1. Depending on the action, it can be triggered when on a press (1) or release (0). The former is most common. Binary continuous are actions that trigger a behavior that continues while the input is held and stops when it is released.

Axis actions are actions that can receive a value between -1 and 1. Axis- is -1 to 0, and Axis+ is 0 to 1.

Two-sided axis inputs, such as the joystick or gamepad axis and mouse motion, are less suited to binary actions because one side would be disabled and most would require strong movement to register as a binary output.

Key and button inputs can be mapped to different areas of axis actions using value transforms. For example, you could map a button to "yaw" (human) with an "invert" value transform and a scale of "0.5" to slowly turn left.

Table 28. Actions

Group	Action	Type	Notes
action-menu			
	close-menu	Binary	
	next-item	Binary	
	page-down	Binary	
	page-up	Binary	
	previous-item	Binary	
	return	Binary	
	select	Binary	
app			
	escape	Binary	
	help	Binary	
	resetView	Binary	Non-standard action name
	toggle-application-mode	Binary	
	toggle-full-screen	Binary	
	user-guide	Binary	

Table 28. Actions (continued)

Group	Action	Type	Notes
binoculars			
	zoom	Axis	
	zoom-in	Axis+	
	zoom-out	Axis-	
combat			
	aim	Binary continuous	
	cycle-weapon-state	Binary	
	fire	Binary continuous	
	next-weapon	Binary	
	reload	Binary	
	select-weapon-1	Binary	
	select-weapon-2	Binary	
	select-weapon-3	Binary	
	select-weapon-4	Binary	
	select-weapon-5	Binary	
	select-weapon-6	Binary	
	select-weapon-7	Binary	
	select-weapon-8	Binary	
	select-weapon-9	Binary	
	select-weapon-10	Binary	
	toggle-weapon-down	Binary	
	toggle-weapon-stowed	Binary	
	weapon-cycle	Binary	
parachute	parachute-left-toggle	Binary	
	parachute-right-toggle	Binary	
commander-rws			
	commander-override	Binary	

Table 28. Actions (continued)

Group	Action	Type	Notes
	slew-to-cue	Binary	
driving			
	brake	Binary continuous	
	differential-lock	Binary	Only enabled when `hasDifferentialLock=true`
	engine-enable	Binary	
	gear-down	Binary	
	gear-up	Binary	
	headlights	Binary	
	parking-brake	Binary	
	pivot	Binary	Only enabled when `canPivot=true`
	steering	Axis	
	steer-left	Axis-	
	steer-right	Axis+	
	throttle	Axis+	
drone			
	camera-elevation	Binary continuous	
	capture-image	Binary	
	drone-collective	Axis	
	drone-pedal	Axis	
	drone-pitch	Axis	
	drone-roll	Axis	
	drone-wheel-brake	Binary	
	toggle-focus-on-drone	Binary	
	toggle-pid-menu	Binary	
	toggle-remote-view-viewport	Binary	
engagement			
	commander-override	Binary	
	fire	Binary continuous	

Table 28. Actions (continued)

Group	Action	Type	Notes
	next-sensor	Binary	
	next-weapon	Binary	
	previous-weapon	Binary	
	slew-to-cue	Binary	
	turret-elevation	Axis	
	turret-azimuth	Axis	
	turret-down	Axis-	
	turret-left	Axis-	
	turret-right	Axis+	
	turret-up	Axis+	
	zoom	Axis	
	zoom-in	Axis+	
	zoom-out	Axis-	
entity-control			
	toggle-action-menu	Binary	
	toggle-debug-menu	Binary	
	reset-vr-head-position	Binary	
	reset	Binary	
	reset-in-place	Binary	
	push-to-talk	Binary continuous	
	toggle-mouse-control	Binary	Only when "mouseControlSupport" is true and the player is not a spectator.
	enable-mouse-control	Binary	
extended-flight-controls			
	autopilot-mode-1	Binary	
	autopilot-mode-2	Binary	
	autopilot-toggle	Binary	
	cycle-weapon	Binary	
	flap1	Binary	
	flap2	Binary	

Table 28. Actions (continued)

Group	Action	Type	Notes
	flap-toggle	Binary	
	launch-chaff	Binary	
	launch-flares	Binary	
	nose-wheel-steering	Binary continuous	
	replenish-fuel-and-ammo	Binary	
	suppress-warnings	Binary	
	toggle-ar-camera-menu	Binary	
	toggle-countermeasures	Binary	
	toggle-landing-gear	Binary	
	toggle-observer-attach-mode	Binary	
	trigger	Binary continuous	
fixed-wing			
	air-brake	Axis	RT Dynamics only
	flap-state	Axis	RT Dynamics only
	landing-gear	Axis	RT Dynamics only
	pedal	Axis	
	pitch	Axis	
	roll	Axis	
	throttle	Axis	
	wheel-brake	Axis	RT Dynamics only
general (commander)			
	sensor-azimuth	Axis	
	sensor-elevation	Axis	
	sensor-zoom-in	Axis+	
	sensor-zoom-in	Axis	
	sensor-zoom-out	Axis-	
general (human)			
	first-person-mode	Binary	

Table 28. Actions (continued)

Group	Action	Type	Notes
	free-look	Binary continuous	
	minimap-cycle	Binary	
	minimap-reset-zoom	Binary	
	minimap-zoom-in	Axis+	
	minimap-zoom-out	Axis-	
	mouse-pitch	Axis	Mouse only
	mouse-yaw	Axis	Mouse only
	move	Axis	
	move-backward	Axis-	
	move-forward	Axis+	
	parachute-deploy	Binary	Loads parachute group
	pitch	Axis	
	pitch-down	Axis-	
	pitch-up	Axis+	
	posture-down	Binary	
	posture-prone	Binary	
	posture-select	Binary	
	posture-stand	Binary	
	posture-up	Binary	
	sprint	Binary continuous	
	strafe	Axis	
	strafe-left	Axis-	
	strafe-right	Axis+	
	third-person-mode	Binary	
	toggle-ar-camera-menu	Binary	
	toggle-binoculars	Binary	
	toggle-compass	Binary	
	toggle-compass-units	Binary	
	toggle-flashlight	Binary	

Table 28. Actions (continued)

Group	Action	Type	Notes
	toggle-nvg	Binary	
	toggle-view-mode	Binary	
	walk	Binary continuous	
	yaw	Axis	
	yaw-left	Axis-	
	yaw-right	Axis+	
general (sensor operator)			
	capture-image	Binary	
	cycle-sensor	Binary	
	cycle-weapon	Binary	
	fire-weapon	Binary continuous	
	sensor-azimuth	Axis	
	sensor-elevation	Axis	
	sensor-reset	Binary	
	sensor-track-location	Binary	
	sensor-track-object	Binary	
	sensor-zoom	Axis	
instruments			
	clear-targets	Binary	
	cop-cursor-x	Axis	
	cop-cursor-y	Axis	
	cycle-targets-backward	Binary	
	cycle-targets-forward	Binary	
	dms-down	Binary	
	dms-left	Binary	
	dms-right	Binary	
	dms-up	Binary	
	radar-cursor-deselect	Binary	

Table 28. Actions (continued)

Group	Action	Type	Notes
	radar-cursor-down	Axis-	
	radar-cursor-left	Axis-	
	radar-cursor-right	Axis+	
	radar-cursor-select	Binary	
	radar-cursor-up	Axis+	
	radar-cursor-x	Axis	
	radar-cursor-y	Axis	
	sar-request	Binary	
	sensor-azimuth	Axis	
	sensor-cycle	Binary	
	sensor-elevation	Axis	
	sensor-reset	Binary	
	sensor-track-location	Binary	
	sensor-track-object	Binary	
	sensor-zoom	Axis	
	toggle-rwr-sounds	Binary	
laser-target-designator			
	press-and-hold-to-lase	Binary continuous	
	toggle-laser	Binary	
rotary-wing			
	collective	Axis	
	pedal	Axis	RT Dynamics only
	pitch	Axis	
	roll	Axis	
	wheel-brake	Axis	RT Dynamics only
	yaw	Axis	non-RT Dynamics only
spectator			
	azimuth	Axis	
	elevation	Axis	
	reset-view	Binary	

Table 28. Actions (continued)

Group	Action	Type	Notes
	zoom	Axis	
	zoom-in	Axis+	
	zoom-out	Axis-	
view			
	next-view	Binary	
	previous-view	Binary	
	view-1	Binary	
	view-2	Binary	
	view-3	Binary	
	view-4	Binary	
	view-5	Binary	
	view-6	Binary	
	view-7	Binary	
	view-8	Binary	
	view-9	Binary	
	view-10	Binary	



12. Controllable Entities

This chapter lists the VR-Forces entity types that are controllable from VR-Engage out of the box. You can configure other entities to be controllable, as described in "Chapter 13. Adding and Editing VR-Engage Entity Models" on page 275.

12.1. VR-Forces Entities that VR-Engage Can Control 240

12.1. VR-Forces Entities that VR-Engage Can Control

The VR-Engage SMS includes the following entities that you can control in VR-Forces from VR-Engage if VR-Forces is running a scenario that uses the VR-Engage SMS. Each entity lists the roles that are available for that entity.

- AH-64D Apache Longbow: Pilot, Sensor Operator



- AH-64E Apache Guardian with FCR: Pilot, Sensor Operator



- Afghan Police Officer: Human Player



- Afghan Soldier: Human Player



- African Insurgent: Human Player



- Agent M9: Human Player



- Ambulance (Vortex): Driver



- AN/PED-1 LLDR: Operator Lasing



- Australian Soldier EF88: Human Player



- Australian Soldier XM107: Human Player



- Beechcraft MC-12 King Air Single Seater: Pilot, Sensor Operator



- Beechcraft MC-12 King Air: Pilot, Sensor Operator



- Brazilian Soldier: Human Player



- CH-149 Cormorant: Pilot, Sensor Operator



- CH-47 Chinook: Pilot, Sling Load Operator



- Chilean M16-M203: Human Player



- Chilean M16: Human Player



- Chilean M4: Human Player



- Civilian Female With Drone: Human Player



- Civilian Male With Drone: Human Player



- Colombian Police Officer: Human Player



- Colombian Soldier: Human Player



- DI AK-47: Human Player



- DI Lasing (CIS): Human Player



- DI Lasing (US): Human Player



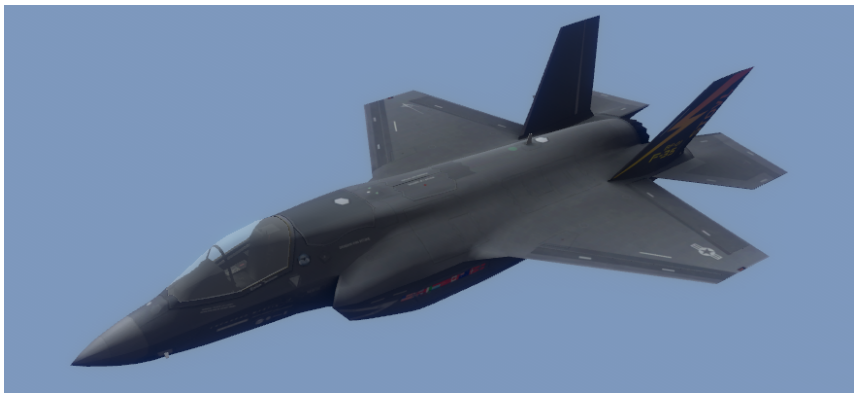
- Eurofighter Typhoon: Pilot, Sensor Operator



- F-16A Fighting Falcon: Pilot, Sensor Operator



- F-35 Lightning II: Pilot, Sensor Operator



- F/A-18 Hornet: Pilot, Sensor Operator



- Farmer M9: Human Player



- Fiat 124 (Vortex): Driver



- GTK Boxer APC: Driver, Gunner, Driver/Gunner



- GTK Boxer CRV: Driver, Gunner, Driver/Gunner, Commander RWS, Commander Hatch



- HMMWV Utility Vehicle: Driver



- ISIS Fighter: Human Player



- Insurgent M9: Human Player



- Insurgent: Human Player



- Iranian Soldier: Human Player



- Iraqi Police Officer: Human Player



- Iraqi Soldier: Human Player



- Israeli Soldier M4: Human Player



- Israeli Soldier: Human Player



- Kurdish Fighter: Human Player



- LAV III APC (Vortex): Driver, Gunner, Driver/Gunner, Commander



- Libyan Police Officer: Human Player



- Libyan Soldier: Human Player



- M1120A4 HEMTT LHS: Seat 1 Driver



- M1126 Stryker ICV with M2: Driver, Gunner



- M1151 ECAC with M240: Driver, Gunner



- M1151 ECAC with M240B: Driver, Gunner



- M1151 ECAC with M2: Driver, Gunner



- M1151 ECAC with MK 19: Driver, Gunner



- M1151 Expanded Capacity Armament Carrier (ECAC): Driver



- M1A1 AIM (AUS) Mineplow: Driver, Gunner, Driver/Gunner, Commander Hatch



- M1A1 AIM (AUS): Driver, Gunner, Driver/Gunner, Machine Gun, Commander Hatch, Commander



- M1A1 SA: Driver, Gunner, Driver/Gunner, Machine Gun, Commander, Commander Hatch



- M1A2 Abrams MBT (Vortex): Driver, Gunner, Driver/Gunner, Machine Gun, Commander, Commander Hatch



- M1A2 SEP Abrams: Driver, Gunner, Driver/Gunner, Machine Gun, Commander, Commander Hatch



- M2A3 Bradley IFV: Driver, Gunner, Commander



- M915 Tractor Truck: Driver



- M977 HEMTT Cargo Truck: Driver



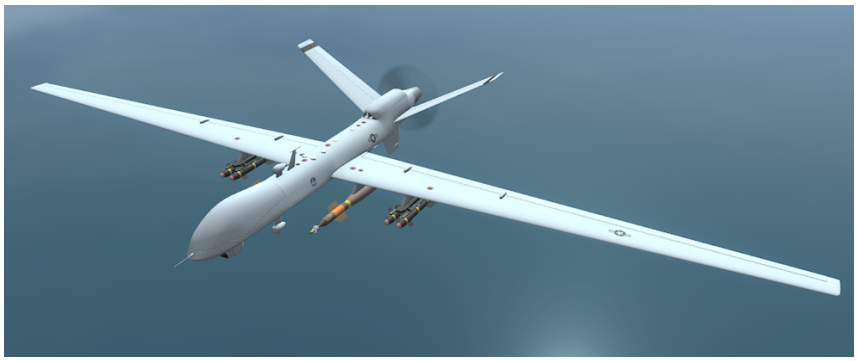
- MH-60 Black Hawk: Pilot, Sensor Operator



- MQ-8B Fire Scout: Sensor Operator



- MQ-9 Reaper UAV: Sensor Operator



- MiG-29 Fulcrum: Pilot, Sensor Operator



- Mideast Combatant: Human Player



- Pakistani Police Officer: Human Player



- Pakistani Soldier: Human Player



- Paratrooper M9: Human Player



- Paratrooper: Human Player



- Police Cruiser (Vortex): Driver



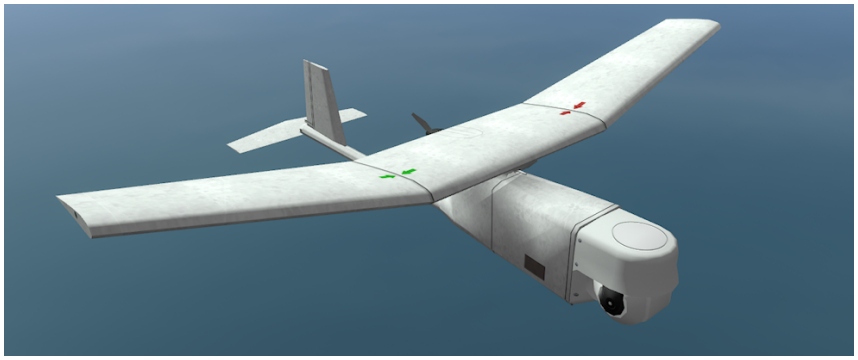
- Police Officer With Drone: Human Player



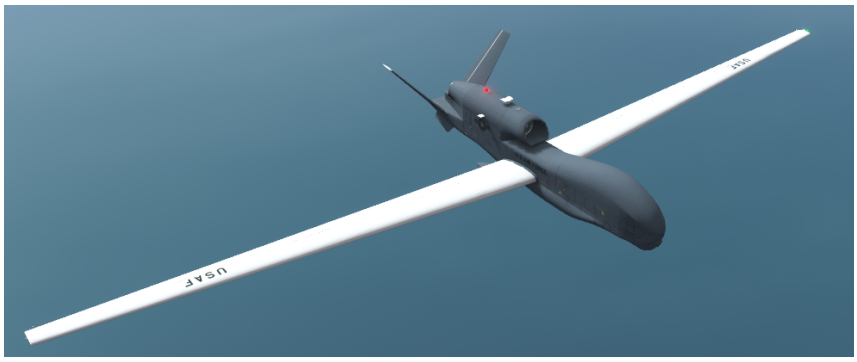
- Policeman M9: Human Player



- RQ-11 Raven UAV: Sensor Operator



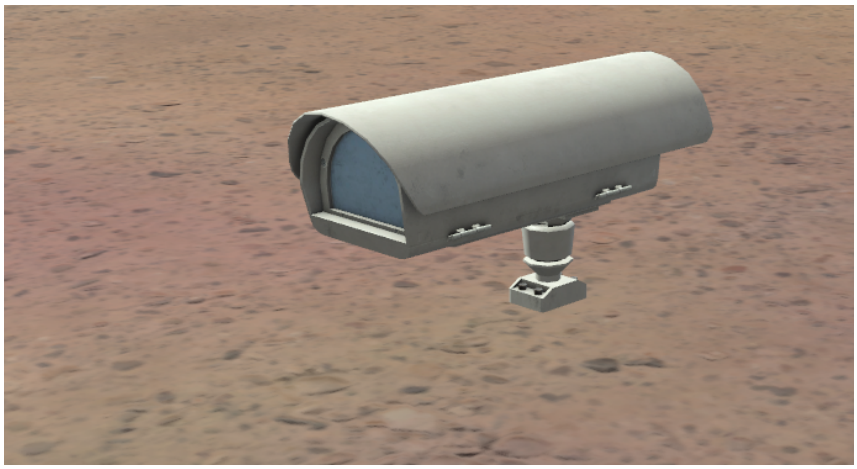
- RQ-4 Global Hawk UAV: Sensor Operator



- RQ-7 Shadow UAV: Sensor Operator



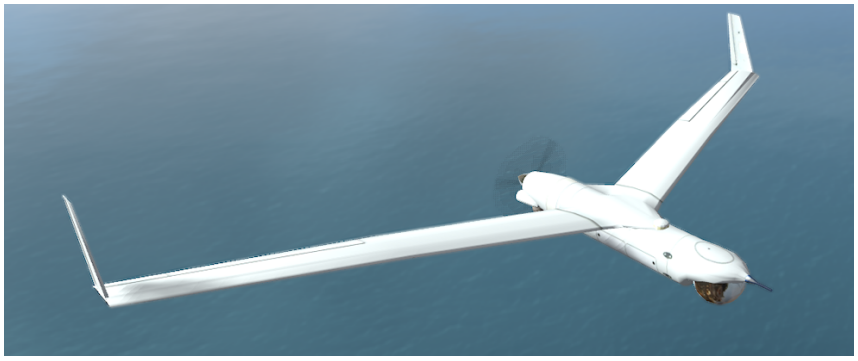
- Remote Camera: Sensor Operator



- Russian Rebels: Human Player



- ScanEagle UAV: Sensor Operator



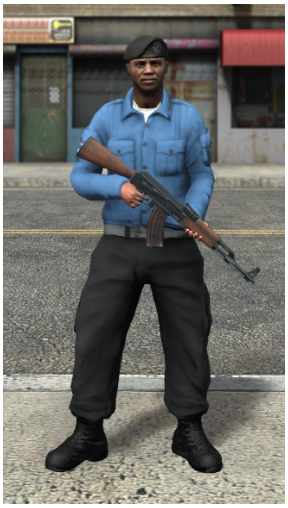
- Security Guard M9: Human Player



- Singaporean Soldier: Human Player



- Somali Police Officer: Human Player



- Somali Soldier: Human Player



- Sport Hatchback Coupe: Driver



- Sport Standard Size Sedan: Driver



- Standard MidSize Sedan: Driver



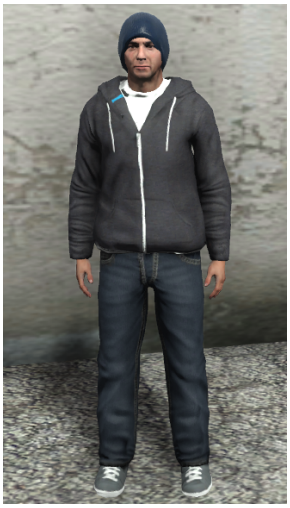
- Sudanese Police Officer: Human Player



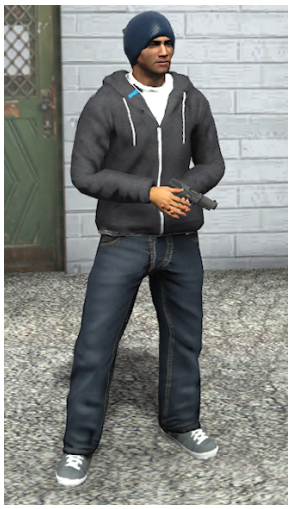
- Sudanese Soldier: Human Player



- Suspect: Human Player



- Suspect M9: Human Player



- Syrian Police Officer: Human Player



- Syrian Soldier: Human Player



- T-38C Talon: Pilot, Sensor Operator



- Taliban: Human Player



- Technical Truck: Driver



- UH-60M Blackhawk: Pilot



- US Army Javelin (Tripod): Human Player



- US Army Javelin: Human Player



- US Army M16-M203: Human Player



- US Army M16: Human Player



- US Army M4: Human Player



- US Army M9: Human Player



- US Border Patrol M9: Human Player



- USMC M16-M203: Human Player



- USMC M16: Human Player



- USMC M4: Human Player



- Ugandan Police Officer: Human Player



- Ugandan Soldier: Human Player





13. Adding and Editing VR-Engage Entity Models

This chapter explains how to edit the VR-Engage simulation model set to add or edit models that you can control from VR-Engage. You can also read "Chapter 14. Model Customization Tutorial" on page 287 for an example.

13.1. The VR-Engage Simulation Model Set 276
13.2. Ground Vehicles and Vortex 276
13.3. Extend the VR-Engage SMS 278
13.3.1 Creating a New Simulation Model Set 278
13.3.2 Creating a New VR-Engage-Controllable Entity 279
13.3.3 Editing a VR-Engage-Controllable Entity 280
13.3.4 Deploying the SMS to VR-Engage 280
13.4. VR-Engage Extensions to .entity Files 281
13.4.1 Structure of Role Data in .entity Files 281
13.4.2 The vrEngageRoles and role Entries 281
13.4.3 The parameterOverride Entry 283
13.4.4 Role-Slot Association 284
13.4.5 Roles Included with VR-Engage 284

13.1. The VR-Engage Simulation Model Set

The entity models that VR-Engage simulates are defined in a set of files called a simulation model set (SMS). VR-Engage includes an SMS called *VR-Engage.sms*. The example scenarios included with VR-Engage all use this SMS. If you create an unhosted session, the list of entities available to you is drawn from *VR-Engage.sms*. If you install the VR-Forces Add-on package, *VR-Engage.sms* is installed into VR-Forces.

VR-Engage.sms includes all of the simulation objects in the default VR-Forces SMS, *EntityLevel.sms*. A subset of human, ground, and air entities are modified so that they can be controlled by a VR-Engage player. When you create a scenario in VR-Forces for use with VR-Engage, you can use any of the simulation objects in the SMS. The extended *VR-Engage.sms* entities function in the scenario as they normally would (that is, as they do in *EntityLevel.sms*) when they are controlled by VR-Forces. However they can also be controlled at any time by a VR-Engage user.

VR-Engage ships with a set of entities that you can control, listed in "Chapter 12. Controllable Entities" on page 239. You can edit the entity models in *VR-Engage.sms* and add new ones. You add and edit models using the VR-Forces Simulation Object Editor, an application included with VR-Forces. You can then use your SMS in scenarios that you create with VR-Forces, as described in "Chapter 15. Creating VR-Engage Scenarios in VR-Forces" on page 297. If you do not have VR-Forces, contact your MAK Technologies salesperson for information about obtaining VR-Forces.



This chapter assumes that you understand the concepts of SMSs and how to use the Simulation Object Editor. For details, please see the relevant chapters in Section XI, “Creating and Editing Simulation Models,” in the *VR-Forces Users Guide*.

13.2. Ground Vehicles and Vortex

VR-Engage uses the Vortex product from CM Labs to implement high-fidelity vehicle dynamics for ground vehicles when they are controlled by players. When a VR-Engage player station takes control of (engages with) a vehicle in a VR-Forces scenario, the back-end simulation model automatically switches from the standard VR-Forces CGF movement system to one that relies on Vortex. Different entity types can be configured to use different Vortex dynamics models, which are called "mechanisms".

VR-Engage includes a limited number of unique Vortex mechanisms to represent the basic types of vehicles (civilian car, truck, tank, IFV, and so on). Various specific vehicle types in the default SMS (*VR-Engage.sms*) are configured to use whichever of those generic mechanisms is the closest match. This is often sufficient for applications that use VR-Engage as a role player station. If you are using VR-Engage as the basis for a true driving trainer, this typically requires a vehicle-specific mechanism that accurately represents your vehicle's correct geometry and mechanical properties.

Editing or authoring custom Vortex mechanisms for use with VR-Engage requires the Vortex Studio editor, which is a separate product that is available directly from CM Labs. Please contact MAK (sales@mak.com or your account representative) if you have questions about obtaining Vortex Studio.

There are several steps required to author a custom Vortex mechanism and configure a VR-Engage entity type to use it:

1. Use the Vortex editor to author the Vortex mechanism. If you want to start with one of the generic Vortex mechanisms that are shipped with VR-Engage, they can be found in *./data/Vortex/mechanisms*. The Vortex Studio documentation describes the process for editing or authoring mechanisms, but the process includes the ability to import your 3D visual model so that you can configure your Vortex mechanism to perfectly match its collision geometry.
2. Configure a VR-Engage movement "system" to use your custom Vortex mechanism. VR-Engage's human readable "system" files are in *./data/simulationModelSets/VR-Engage/vrfSim/systems*. A system's reference to a specific Vortex mechanism looks like this:

   ```
   (vortex-mechanism-filename "$(DATA_
   DIR)\\Vortex\\mechanisms\\technicalTruck\\Dynamic\\Design\\Truck.v
   xmechanism").
   ```

   Change the reference to point to your desired mechanism. Besides configuring a system to use the correct Vortex mechanism, you can also edit the *sysdef* file to map named parts of the Vortex mechanism to DIS/HLA articulated part and parameter numbers, and to map VR-Engage inputs to Vortex inputs.
3. Use the Simulation Object Editor, included with VR-Forces, to ensure that your entity type is configured to use the movement system that you edited.
4. Finally, in order for your custom mechanisms to run in VR-Engage without requiring a Vortex runtime license, use the Vortex signing tool to sign your content. Run *./bin64/vortex/tools/VortexSigningTool.exe*, then drag and drop the flies you want to sign into the tool. You must sign the mechanisms again each time you edit them.

Because the standard VR-Forces CGF movement system for a ground vehicle is replaced by a higher-fidelity Vortex-based movement system when an entity is player-controlled, certain parameters that you can see and edit in the SOE do not apply when the entity is player-controlled. For example, when the CGF movement system is active, the dimensions of the entity's bounding box dictate when an entity collides with a wall or other obstacle. But when an entity is player-controlled in VR-Engage, the simple bounding box parameters are ignored when computing ownship collisions. Instead, the Vortex-based model computes collisions based on higher-fidelity geometry information that is configured in the Vortex mechanism – including wheel sizes and locations, as well as runtime knowledge about the positions of any articulated parts, shock absorber compressions, gear ratios, automatic transmission parameters, and so on. However, the bounding box dimensions are still used for purposes other than ownship collisions, for example, they are still used by VR-Forces to compute whether the VR-Engage-controlled entity blocks line of sight between VR-Forces CGF entities, or whether a VR-Forces entity's path is blocked by the VR-Engage controlled entity. Therefore, it is important to accurately configure the CGF bounding box information in the SOE even for entities types that are usually player-controlled.

Contact MAK support to learn more about authoring and configuring vehicle-specific Vortex mechanisms for use in VR-Engage: https://jira.mak.com/servicedesk/customer/portal/1.

13.3. Extend the VR-Engage SMS

You may want to edit simulation models or add new ones for any of the following reasons:

- Create an entity similar to an existing one (for example, a new type of car or aircraft).
- Change the performance characteristics of an existing entity.
- Replace the 3D model associated with an entity.
- Replace or modify the first person model.

The first three modifications can typically be done using the Simulation Object Editor. Replacing or changing the first person model requires different tools and a different skill set. Contact MAK Technologies at https://jira.mak.com/servicedesk/customer/portal/1 for more information about changing the first person model.

The best practice is to leave the original VR-Engage SMS unmodified and create an extended one. It makes maintaining your changes easier in future version and makes it less likely that you will inadvertently modify an existing entity or behavior.

The overall approach to extending *VR-Engage.sms* is:

1. Create a new SMS that includes *VR-Engage.sms*.
2. Make the desired changes to the SMS.
3. Copy the new SMS to your VR-Engage installation.
4. Configure VR-Engage to use the new SMS by default when starting an unhosted session or creating a new scenario in a hosted session.



This chapter covers some of the basics of working in the SMS, and also a tutorial. For more details, such as how to change the 3D model, refer to the *VR-Forces Users Guide*.

13.3.1 Creating a New Simulation Model Set

If you want to modify or extend the VR-Engage simulation model set, we recommend that you create a new simulation model set that includes *VR-Engage.sms*. This makes the differences between your customizations and the installed SMS clear. It also makes it easier to revert to the installed SMS if your new SMS has problems.

For information about other ways to create an SMS, please see the *VR-Forces Users Guide*.

➤ **To create a new simulation model set:**

1. Open the Simulation Object Editor.

2. Choose **File > New Simulation Model Set**. The New Simulation Model Set dialog box opens.
3. Select New Simulation Model Set.
4. Click Next. The Simulation Model Set Type page displays.
5. Select Extend Existing Simulation Model Set.
6. Click Next. The Extend Existing Simulation Model Set page displays.
7. Select *VR-Engage.sms*.
8. Click Next. The New Model Set Name page displays.
9. In the File Name box, enter a name for the new model set.
10. Click Finish. The Simulation Object Editor loads systems information from the included model sets and displays the new SMS. This can take a few minutes.

    Since all of the simulation objects are included from other SMSs, none of them are currently editable. They are listed in the Objects List using gray text.
11. Choose **File > Save Simulation Model Set**.

13.3.2 Creating a New VR-Engage-Controllable Entity

The recommended way to create a new entity that can be controlled by a VR-Engage player is to base it on an existing entity of the type you want to create. To control an entity in VR-Engage, it must be one of the entities that was included from *VR-Engage.sms*. Entities that are included from *EntityLevel.sms* cannot be controlled from VR-Engage.

When creating a new entity based on an existing one, you can either Inherit or Copy. If you choose Copy, there is no connection to the original entity. Any changes to the original will not affect your new entity. If you choose Inherit, there is a parent-child relationship between your entity and the original. Use Inherit if your model is similar to the parent because it is easier when you upgrade the SMS for the next software version if there are changes to the parent model. Use Copy only when you intend to change everything about the new entity: model, systems, embarkation, and so on.

➤ **To create a new entity that can be controlled by a VR-Engage player:**

1. In the Simulation Object Editor, load your new SMS.
2. In the Tags list, select VR-Engage. The Objects List is filtered to show only models that are included from *VR-Engage.sms*.
3. Select an entity similar to the one you want to create. (A yellow bar is added to the window. It displays the SMS where the selected entity is originally included.)
4. Choose **Model > New Model from Existing (Inherit)** (or **(Copy)**). The Create New Model dialog box opens.
5. Enter a name for the new model.
6. Optionally, edit the Entity Type.
7. Click OK.

8. Edit the new entity.
9. Save the SMS.

13.3.3 Editing a VR-Engage-Controllable Entity

You can only edit models that are part of the loaded SMS. You cannot edit models that are included from another SMS. Therefore, if you want to edit one of the models included from *VR-Engage.sms*, you must promote it to your SMS. (You can always edit any of the models you created using the procedure in "13.3.2 Creating a New VR-Engage-Controllable Entity" on the previous page.)

1. In the Simulation Object Editor, load your SMS.
2. In the Tags list, select VR-Engage. The Objects List is filtered to show only models that are included from *VR-Engage.sms*.
3. Select the entity you want to edit. If the entity is not part of the main SMS, a yellow bar is added to the window. It displays the SMS where the selected entity is currently included.
4. If the entity is not part of the main SMS, choose **Model > Promote to Main Simulation Model Set** or click the Promote button in the yellow bar. The model is promoted. It is now shown in dark text at the top of the Objects List.
5. Edit the model as desired.
6. Save the SMS.

13.3.4 Deploying the SMS to VR-Engage

After you create the new SMS, you must copy it to VR-Engage. This allows you to run scenarios that were created in VR-Forces using the new SMS. You can also specify that the new SMS be used by default when you create an unhosted session or create a new scenario in a hosted session in VR-Engage.

1. Copy the SMS file and associated data folder to your VR-Engage installation, preserving the path. The path to your SMS will look something like this:

   ./data/simulationModelSets/<your SMS name>.sms

   ./data/simulationModelSets/<your SMS name>

2. Configure VR-Engage to use the new SMS by default:
   a. In a text editor, open *./appData/scripts/playerStation.lua*.
   b. Set the variable **defaultSimulationModelSets** to point to your new SMS, for example:

```
  -- Default simulation model set(s) to use when launching the
simulation engine.
  -- Specify list of SMSs separated by commas.
  defaultSimulationModelSets = {
    "$(DATA_DIR)/simulationModelSets/myVR-Engage.sms"
  };
```

13.4. VR-Engage Extensions to *.entity* Files

13.4.1 Structure of Role Data in *.entity* Files

Roles are defined in Lua role definition files and VR-Engage includes a number of role configuration files, which are listed in Table 31 on page 285. Those roles are then referenced in *.entity* files in the `vrEngageRoles` entry, which is placed near the end of the *.entity* file before the `</simObject>` close tag.

```
<simObjects>
  <simObject>
    ...
    <vrEngageRoles>
      <role> ... </role>
      <role> ... </role>
    </vrEngageRoles>
  </simObject>
</simObjects>
```

13.4.2 The `vrEngageRoles` and `role` Entries

The <vrEngageRoles> entry in *.entity* files contains a list of one or more <role> entry for the entity. The <role> entry has two required parameters: paramName and defaultDisplayLayout. There are other, optional parameters.

```
  <role paramName="Driver" slotName="Driver" unlisted="false"
defaultDisplayLayout="1 Screen Horizontal">@(DATA_
DIR)\simulationModelSets\VR-Engage\roles\<role file>.lua
    <parameterOverride> ... </parameterOverride>
      ...
    <parameterOverride> ... </parameterOverride>
  </role>
```

Table 29. role parameters

Type	Name	Default	Required	Description
Attribute	defaultDisplayLayout	1 Screen Horizontal	Yes	The default visual layout for this entity when in that role. A role can include one or more display layouts, depending on the hardware and visual appearance desired. You can refer to an existing role file for the list of available layouts. The role definition file must include at least one display layout set for the default. All roles included in VR-Engage have a `"1 Screen Horizontal"` layout, at the minimum. Each role file includes the `displayLayouts` section at the top. That section defines the components and parameters needed to control the viewpoint and any HUD overlays.
Attribute	paramName		Yes	Name of the role as it will display in the Choose Role panel's Role list.

Table 29. role parameters (continued)

Type	Name	Default	Required	Description
Element	parameterOverride		No	You can specify zero or more parameterOverride child elements. A parameterOverride contains Lua code snippets to override specific table elements in the role configuration file.
Element Text			Yes	Path to the Lua script that defines the role configuration.
Attribute	slotName		No	Name of the Embarkation Slot associated with this role. This role is automatically engaged when the player's entity is embarked in this slot. If this attribute is not provided, this role will not be accessible via embarkation and can only be engaged from the Choose Role panel. For more information about roles and slots, see "13.4.4 Role-Slot Association" on the next page.
Attribute	unlisted	false	No	Boolean indicating whether the role should be available for selection in the Choose Role panel.

13.4.3 The parameterOverride Entry

You can override any parameters from the role configuration file using overrides in the *.entity* file. This allows you to use a single role configuration file in multiple entities with entity-specific customizations made in the appropriate *.entity* files. The

parameters you can override are the entries in the role's Lua `components` table. The syntax for the overrides is: `<parameterOverride paramName="`*`key in components table.lua table entry`*`">`*`value`*`</parameterOverride>`. For example:

```
  <parameterOverride paramName="savedViewUpdater.initialView">"4 -
Driver View"</parameterOverride>
```

Table 30. parameterOverride parameters

Type	Name	Default	Required	Description
Attribute	paramName		Yes	Name of a Lua component table item to override in the role file. The format of this string is <componentName>.<Lua table entry>. In the above example, the `initialView` table member of the `savedViewUpdater` component is the item being replaced.
Element Text			Yes	A Lua code snippet to replace the corresponding entry in the role configuration file.



For an example of editing these values to configure your own customized entity, see "Chapter 14. Model Customization Tutorial" on page 287.

13.4.4 Role-Slot Association

Roles are associated with slots by the `slotName` parameter of the VR-Engage `<role>` entry. The role is then associated with the embarkation slot with a matching `slotName` parameter. A role can only be associated with one slot, and a slot should only be associated with one role. It is not required that every role be associated with an embarkation slot, nor is it required that every slot have an associated role.

If a slot has an associated role, that vehicle role is automatically engaged whenever a player's entity is embarked into that slot, or when a player engages with an entity already embarked in the slot. If a player entity is embarked in a slot that does not have an associated role, the player remains engaged in the embarked human entity.

If a role does not define a valid slot association, there is no way to engage in that vehicle role via embarkation. To play this role, a player must engage directly with the vehicle from the Choose Role panel, or by command from the VR-Engage Manager Panel in VR-Forces.

To learn more about roles and slots, see "Appendix B. Embarkation Slot Configuration in VR-Engage" on page 317.

13.4.5 Roles Included with VR-Engage

Table 31 lists the out-of-box roles configured on entities in *VR-Engage.sms*. Some role files are derived from other, more generalized roles. Many entities in the VR-Engage SMS are configured with more than one role.

Table 31. Available roles

Role Configuration File		Description
airSensorOperator.lua		Aircraft-mounted gimbaled sensor
cameraSensorOperator.lua		Gimbaled sensor such as a security camera
commander.lua		Ground vehicle commander with out-the-hatch viewpoint
	commanderViews.lua	Commander view through ring of periscopes.
diLasingHuman.lua		Human configured with a hand-held laser designator
driver.lua		Ground vehicle driving
	boxerDriver.lua	GTK Boxer vehicle
	lavDriver.lua	LAV vehicle
	tankDriverWith3D.lua	Ground vehicle driving plus display configuration for use with 3D models with high detail interiors
	driverGunner.lua	Combination ground vehicle driving and turreted gun
emergencyVehicleDriver.lua		Ground vehicle driving with lights and sirens
fixedWingPilot.lua		Fixed wing aircraft pilot
	t38Pilot.lua	T-38 aircraft
gunner.lua		Turreted gun
	boxerGunner.lua	GTK Boxer
	hmmwvGunner.lua	HMMWV/ECAC
	lavGunner.lua	LAV
	strykerGunner.lua	Stryker
	tankGunner.lua	Tanks with turreted main guns
	tankMachineGunner.lua	Tanks with secondary machine guns
human.lua		Human with weapon, supports desktop and VR
	humanWithoutVr.lua	Similar to human.lua, but without a VR display configuration.
	militaryHuman.lua	Adds the ability to execute a set of military hand gestures to the other features of human.lua
lldr.lua		Lightweight laser rangefinder designator device operator

Table 31. Available roles (continued)

Role Configuration File	Description
notionalPassenger.lua	Passenger on a vehicle
remoteWeaponSystem.lua	Remotely operated weapon system (CROWS), typically controlled by a commander in a vehicle
rotaryWingPilot.lua	Rotary-wing pilot



14. Model Customization Tutorial

There are a number of tasks involved in the process of creating or customizing a entity for your use. This tutorial shows you how to take an existing VR-Forces model and customize it.

14.1. Overview of Customizing a Model 288
14.2. Extending VR-Engage.sms 288
14.3. Editing the Entity in the SOE 289
14.3.1 Promote the Entity to your SMS 289
14.3.2 Review the Entity's Systems 290
14.4. Editing the Tank's .entity File 290
14.4.1 Specify a VR-Engage-Capable Platform 291
14.4.2 Add the VR-Engage Tag 291
14.4.3 Add the Driving Role Component 291
14.5. Test Driving Your Tank 292
14.5.1 Copy the SMS Files 292
14.5.2 Specify Your SMS as the Default 292
14.5.3 Drive Your Tank 293
14.6. Creating the Views for Your Tank 293
14.6.1 Create a New Driver Views File 294
14.6.2 Override Tank Driver View with Your View 294

14.1. Overview of Customizing a Model

This tutorial demonstrates how to add VR-Engage driver and gunner roles to a VR-Forces CGF entity. Specifically, we will edit the existing Leopard 2 Tank to be player controllable in each of these roles. The Leopard tank is a CGF entity located in the VR-Engage SMS, which is an extension of VR-Forces's EntityLevel SMS. Out of the box, it does not have any VR-Engage role player capabilities.

To complete the tutorial, you must have VR-Engage and a compatible version of VR-Forces with the VR-Engage add-ons. See "Installing the VR-Forces Plug-in" on page 38.

In this tutorial, you will:

1. Create a custom SMS that extends the VR-Engage SMS.
2. Promote an entity to your custom SMS.
3. Make the entity drivable by players.
4. Add custom views for the driver role.

These tasks are beyond the scope of the tutorial:

- Configuring a new Vortex model tailored to this vehicle. Instead, we take advantage of the existing Vortex mechanism file defined for tanks because it is a close enough approximation for the Leopard tank. For more about configuring Vortex models, see "13.2. Ground Vehicles and Vortex" on page 276.
- Mapping the wheel and sprocket articulated parts on the 3D model to the simulation model. The Leopard tank we use in this tutorial has already been configured by an artist to include the necessary articulated parts.
- Configuring popup menus that allow human characters to walk up to the vehicle and choose a role.

14.2. Extending *VR-Engage.sms*

First, create a new SMS that extends *VR-Engage.sms*. The new SMS will include all of the existing VR-Engage-controllable entities and then we'll add an entry for our Leopard 2 Tank. We start by extending the SMS in our VR-Forces installation.

➤ **To create an extension of the existing SMS:**

1. Open the Simulation Object Editor, which is part of your VR-Forces installation. In Windows, open the Start menu and then choose **MAK Technologies > Simulation Object Editor**.
2. Choose **File > New Simulation Model Set**. The New Simulation Model Set dialog box opens.
3. Select New Simulation Model Set.
4. Click Next. The Simulation Model Set Type page displays.
5. Select Extend Existing Simulation Model Set.
6. Click Next. The Extend Existing Simulation Model Set page displays.

7. Select *VR-Engage.sms*.
8. Click Next. The New Model Set Name page displays.
9. In the File Name box, enter TankTutorial.
10. Click Finish. The Simulation Object Editor loads systems information from the included model sets and displays the new SMS. This can take a few minutes.

    Since all of the simulation objects are included from other SMSs, none of them are currently editable. They are listed in the Objects List using gray text.

14.3. Editing the Entity in the SOE

There are some edits that you must make in the Simulation Object Editor (SOE).

- We'll change the movement system to the VR-Engage Tracked Vehicle system. That system has the same capabilities of the Tracks system, which MAK applications use when the vehicle is under CGF control. It also has the higher-fidelity Vortex-based movement model when the tank is under player control.
- In the SOE, you see other parameters that are available for configuration in the Movement tab such as mass, max-speed, max-slope, and so on. These only apply to the lower-fidelity VR-Forces CGF movement model. The comparable model parameters for the higher-fidelity Vortex model are in the Vortex mechanism file, separate from the SOE settings.
- In the General tab for the Leopard 2 Tank, you can see that the Platform is set to Ground Vehicle. This is the platform file set for most ground vehicles and contains the components and parameters common to ground vehicle platforms. Like the Movement system we need to change it to a version that extends the entity with additional VR-Engage capabilities. You cannot edit that using the SOE, however, so we will edit it in a text editor later in this tutorial.

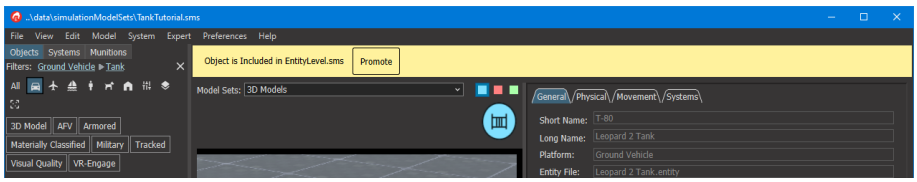
14.3.1 Promote the Entity to your SMS

To make an existing VR-Forces entity part of your custom VR-Engage SMS, you promote it.

➤ **To promote the tank entity:**

1. In the Objects list, select Leopard 2 Tank. Optionally, you can use the filters to narrow the list.
2. In the yellow bar near the top of the window, click Promote (Figure 91). The entity is now available to edit in your SMS.

Figure 91. Promote Simulation Object



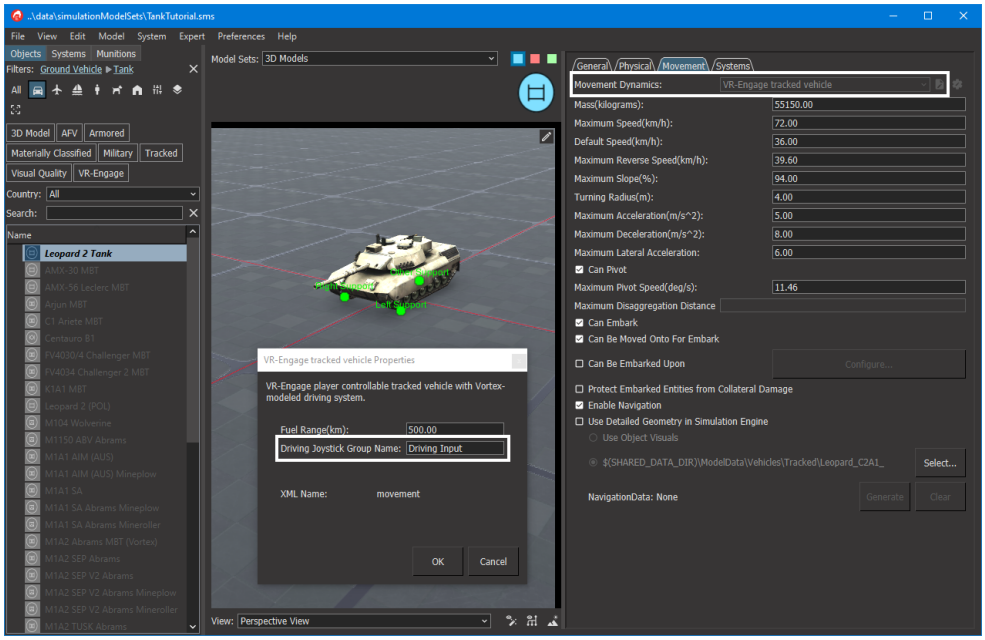
14.3.2 Review the Entity's Systems

To make the Leopard 2 Tank entity drivable as a player-controlled entity, replace the CGF-only movement system with the corresponding VR-Engage movement system. It provides all of the CGF capabilities for autonomous movement, allowing it to be controlled by the CGF in a scenario, and also adds the ability for players to assume control and drive the vehicle themselves.

➤ **To configure the movement system:**

1. Select the Movement tab.
2. From the Movement Dynamics list, choose VR-Engage Tracked Vehicle.
3. Click Edit Properties . The VR-Engage Tracked Vehicle Properties dialog box opens.
4. Note of the value for the Driving Joystick Group Name, which is Driving Input. We use that later in the tutorial when we configure the driving role.
5. Choose **File > Save Simulation Model Set**.
6. Close the SOE. If you are prompted to do so, click OK to save your changes.

Figure 92. Leopard 2 Tank in SOE



14.4. Editing the Tank's *.entity* File

We need to edit the *Leopard 2 Tank.entity* file to configure our tank as an entity that can be controlled and driven by VR-Engage players. We can edit the entity in a text editor and then open it again in the SOE without losing the changes we made.

You can look at another *.entity* file such as *./data/simulationModelSets/VR-Engage/vrfSim/vreM1A2_Abrams_MBT.entity* to see its configuration.

14.4.1 Specify a VR-Engage-Capable Platform

All VR-Engage entities must be configured with a VR-Engage-specific platform file. Platform files contain components and parameters common to all entities of a given category: human, ground vehicle, air vehicle, and so on.

The VR-Engage versions of these platform files extend the corresponding entity-level SMS versions with various overrides related to embarkation, weapon firing, state repository extensions, and more.

For our tank, we'll replace `Ground_Vehicle.ope` with `Vortex_Tracked_Vehicle.ope`. We do this by editing the tank's *.entity* file.

➤ **To open the entity file and edit the platform:**

1. In your VR-Forces installation, navigate to *./data/simulationModelSets/TankTutorial/vrfSim* and then open the *Leopard 2 Tank.entity* file in a text editor.
2. Near the top of the file, edit the simObject entry to change the platform. When you are finished, the line should read:

   ```
   <simObject objectType="1:1:1:78:1:2:0:0"
   matchType="1:1:1:78:1:2:0:0" platform="@(platforms-dir)/Vortex_
   Tracked_Vehicle.ope">
   ```

3. Save the file.

14.4.2 Add the VR-Engage Tag

The `gui-categories` parameter determines the filters available in the Simulation Objects Palette that allow users to filter the list of entities when they are making their selection. Adding the `VR-Engage` tag to this parameter helps VR-Forces users choose VR-Engage-controllable entities when they create a scenario.

➤ **To add the VR-Engage tag:**

1. In the *Leopard 2 Tank.entity* file, locate the `gui-categories` parameter. It's near the bottom of the file.
2. Add `"VR-Engage"` to the list. It should now look something like:

   ```
   <string paramName="gui-categories">"3D Model" "Armored" "Ground
   Vehicle" "Materially Classified" "Military" "Tank" "Tracked"
   "Visual Quality" "VR-Engage"</string>
   ```

3. Save the file.

14.4.3 Add the Driving Role Component

Finally, we add the tank driver role to the *.entity* file. This connects our tank to the driving system (the `Vortex_Tracked_Vehicle.ope` platform) that we specified earlier. The role itself is defined in a role definition file, in this case *tankDriver.lua*. The `<role>` entry in *.entity* files has two parameters, as described in "13.4.2 The vrEngageRoles and role Entries" on page 281.

➤ **To add the driving role:**

1. Locate the <vrEngageRoles/> entry, near the bottom of the *Leopard 2 Tank.entity* file and immediately before the closing tag of </simObject>.
2. Replace the <vrEngageRoles/> entry with:

```
  <vrEngageRoles>
    <role paramName="Driver" defaultDisplayLayout="1 Screen
Horizontal">@(DATA_
DIR)\simulationModelSets\VR-Engage\roles\tankDriver.lua</role>
  </vrEngageRoles>
```

3. Save the file.

14.5. Test Driving Your Tank

To test drive your tank to make sure that it is VR-Engage-controllable, first we need to copy the SMS files from the VR-Forces directory to the VR-Engage directory. We also need to update the *playerStation.lua* file to use the TankTutorial SMS. After we complete those tasks, we can use an unhosted session of VR-Engage to create our entity and drive it.

14.5.1 Copy the SMS Files

➤ **To copy the SMS files:**

1. Navigate to your VR-Forces installation and the *./data/simulationModelSets* directory.
2. Select the *TankTutorial.sms* file and the *TankTutorial* directory.
3. Copy these files.
4. Navigate to your VR-Engage installation and the *./data/simulationModelSets* directory.
5. Paste the files.

14.5.2 Specify Your SMS as the Default

While VR-Engage supports multiple SMSs, we do not need to use more than the one you created earlier. It includes everything from the default *VR-Engage.sms file*, plus your Leopard 2 Tank.

➤ **To specify that VR-Engage should use your SMS:**

1. Navigate to *./appData/scripts* and then open the *playerStation.lua* file in a text editor.
2. Near the top of the file, edit the defaultSimulationModelSets parameter to use your custom SMS. It should now look like:

```
  -- Default simulation model set(s) to use when launching the
simulation engine.
  -- Specify list of SMSs separated by commas.
  defaultSimulationModelSets = {
```

```
        "$(DATA_DIR)/simulationModelSets/TankTutorial.sms"
    };
```

3. Save the file.

> The `$(DATA_DIR)` variable points to the *./data* directory in your VR-Engage installation.

14.5.3 Drive Your Tank

We will start an unhosted scenario in VR-Engage that will use your SMS. (You cannot use any scenarios included with VR-Engage to place and drive your tank because they all use the *VR-Engage.sms*.)

➤ **To start an unhosted session and then drive your tank:**

1. On the Start menu, choose **MAK Technologies > VR-Engage**. The VR-Engage window opens.
2. In the Choose Connection screen, select the connection that you want to use and then click Connect. DIS (7) Localhost is sufficient for a simple test.
3. In the Choose Session screen, click Start Unhosted Session.
4. In the Choose Terrain list, select Ground_DB II.mtf.
5. Click Load Terrain. VR-Engage starts the backend and loads the terrain.
6. In the Choose Role panel, select Create a New Entity.
7. Select your Leopard 2 Tank and then click on the terrain to place it.
8. Click Engage.
9. Click the Play button  to start the scenario.
10. Drive the tank. See "6.2. Driving Ground Vehicles" on page 137 to learn more about operating tanks.
11. You can change the viewpoint as described in "4.3. Changing the View Mode" on page 101.

    These viewpoints were defined for another vehicle. We'll customize the viewpoints in the next sections of this tutorial.

14.6. Creating the Views for Your Tank

If you changed your viewpoint in the previous task, you may have noticed that one of the views is close to the driver's perspective but was partially obscured by the vehicle. That viewpoint was configured for a different vehicle, the M1A2 Abrams. The geometry of the Leopard 2 Tank's 3D model is slightly different and we want to create a new driver view for our tank, which we do by creating an observer views file.

14.6.1 Create a New Driver Views File

This procedure assumes that you have completed the tasks in "14.5. Test Driving Your Tank" on page 292 and currently have the Leopard 2 Tank in an unhosted session.

➤ **To create the observer views file:**

1. If you are in the driver role of Leopard 2 Tank, press **Esc** and then click Choose Role.
2. Press **Shift**+**F12** to switch to Admin mode.
3. Choose **View > Attachments Panel**.
4. Choose **View > Observer Views Panel**.
5. Optionally, undock these two panels and move them somewhere that you can see them clearly.
6. Press the Spacebar to position the observer directly behind the entity. Your tank's marking text now displays in the Attached Objects list in the Attachments Panel.
7. In the Observer Views panel, click Clear Observer Views (). You want to start with a blank slate of viewpoints.
8. Click Yes in the confirmation message.
9. In the Attachment Panel, change the attachment mode from Follow to Mimic.
10. Using the standard keyboard controls for the observer, position the viewpoint where you want it to be.
11. In the Observer Views panel, click Add Observer View (). It is added to the list with a default name.
12. Optionally, you can edit the name. Double-click to edit the name and then press **Tab** when you are finished, or click on the VR-Engage title bar. For this tutorial, we refer to this viewpoint as "Driver View".
13. In the Observer Views Panel, click Export All Views to a File ().
14. In the confirmation message, click Save As. The Save Observer File As dialog box opens.
15. Enter in the File Name leopardSavedViews.
16. Click Save. The file is saved in *./appData/settings/vrEngage*.

14.6.2 Override Tank Driver View with Your View

For this tutorial, we will not create a new role or edit the existing *tankDriver.lua* file. Instead, we'll edit the entity file, referencing our saved views file and setting our view as the initial view.

We'll use overrides in the *.entity* file to override the saved views. To learn more about overrides, see "13.4.3 The parameterOverride Entry" on page 283.

For example, if you look at the *./data/simulationModelSets/VR-Engage/roles/tankDriver.lua* file, you can see the saved views and initial views settings that we are going to override. They are in the DtSavedViewUpdater component (lines 24 and 25).

```
21    ["savedViewUpdater"] = {
```

```
    componentType = "DtSavedViewUpdater";
    priority = 10;
    savedViews="m1DriverViews.osrx";
    initialView="1 - Driver";
    allowChangingViews = true;
    viewChannelKeywordMap={
    };
    clippingPlaneSettings = {
      ["default"] = {
        { mode = "default"; nearClipDistance = 0.25; clipPolicy =
0; dynamicNearClipAttachedObjectPolicy = 0; }
      };
    };
  };
```

➤ **To add your viewpoint to the *.entity* file and override the role settings:**

1. In your VR-Engage installation, open the *./data/simulationModelSets/TankTutorial/vrfSim/Leopard 2 Tank.entity* file in a text editor.
2. Edit the driver role entry to add parameter overrides to use your views file and driver view:

```
  <vrEngageRoles>
    <role paramName="Driver" defaultDisplayLayout="1 Screen
Horizontal">@(DATA_DIR)\simulationModelSets\VR-
Engage\roles\tankDriver.lua
      <parameterOverride
paramName="savedViewUpdater.savedViews">"leopardSavedViews.osrx"</
parameterOverride>
      <parameterOverride
paramName="savedViewUpdater.initialView">"Driver
View"</parameterOverride>
    </role>
  </vrEngageRoles>
```

3. Save the file.

To test and see your changes, you must close and restart VR-Engage to allow it to pick up your parameter overrides.



15. Creating VR-Engage Scenarios in VR-Forces

This chapter describes how you can create your own scenarios that include VR-Engage controllable entities. While VR-Engage can load or join a session of any entity-level VR-Forces scenario, players can only choose an entity that is configured as a VR-Engage controllable entity. VR-Engage does not allow you to create and save scenarios. You need to use VR-Forces with the VR-Engage add-ons to create scenarios that include VR-Engage controllable entities.

15.1. Installing the VR-Forces/VR-Engage Add-ons 298
15.2. Creating a VR-Engage Scenario in VR-Forces 298
15.3. Running Your Scenario 300
15.3.1 Hosting a Scenario in VR-Forces 300
15.3.2 Hosting a Scenario in VR-Engage 300

15.1. Installing the VR-Forces/VR-Engage Add-ons

Before you can create your own VR-Engage scenarios, you must install the *VR-ForcesVR-EngageAdd-ons.exe* plug-in on a version of VR-Forces that is compatible with your copy of VR-Engage. In addition to other features, the installer extended VR-Forces with the *VR-Engage.SMS*, which includes entities that a VR-Engage user can control. These entities also have all of the built-in capabilities of the corresponding entity from the VR-Forces *EntityLevel.SMS*.

For more information see "Installing the VR-Forces Plug-in" on page 38.

15.2. Creating a VR-Engage Scenario in VR-Forces

You can use VR-Forces to create scenarios. If you want to include entities that can be controlled by VR-Engage players, you must use the *VR-Engage.sms* (or an SMS that is an extension of the VR-Engage SMS) and include entities that have been configured as controllable entities. "Chapter 12. Controllable Entities" on page 239 lists the entities that are included with VR-Engage. "Chapter 13. Adding and Editing VR-Engage Entity Models" on page 275 explains how to configure other models to be controlled by VR-Engage players. When you use the VR-Engage filter in the Simulation Objects Palette (Figure 94), you see a list of VR-Engage controllable entities.

Your scenario can include other entities that are not VR-Engage controllable and players can interact with them. The VR-Engage SMS is an extension of VR-Forces's Entity Level SMS, including all of the simulation objects, which allows you to add them to VR-Engage scenarios. (Non-controllable entities are not listed in the Choose Role panel.)

You can assign tasks and plans to all entities in the scenario. If a player engages as a controllable entity, its plan or task is suspended while the player is engaged. That entity then resumes its assigned behaviors when the player disengages. You can also use plan triggers to give entities responses to players' actions.

Before creating the scenario, ensure that you are using a version of VR-Forces that is compatible with the version of VR-Engage that you want to use and also has the add-ons plug-in for that version of VR-Engage. To check for compatibility, see the "New Features and Updates" in the *VR-Engage Release Notes*.

➤ **To create a scenario that is compatible with VR-Engage:**

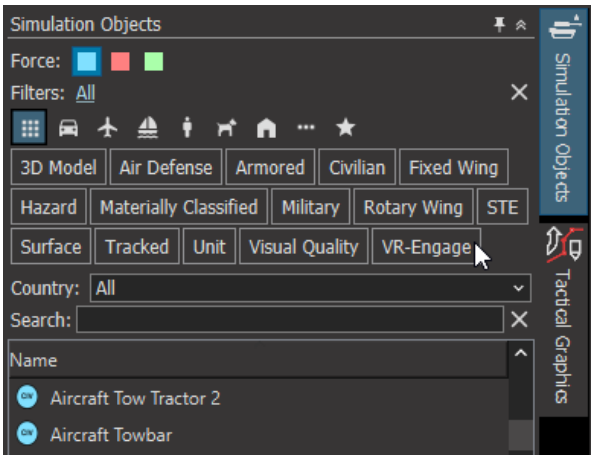
1. Open VR-Forces.
2. In the Scenario Startup dialog box, specify VR-Engage as your Simulation Model Set (Figure 94), or select an SMS that you created.

Figure 93. New scenario in VR-Forces Scenario Startup



3. Click OK.
4. By default, VR-Forces displays the Simulation Objects and Tactical Graphics palettes. You can add any simulation objects or tactical graphics included in the palettes, not only VR-Engage controllable entities.
5. To filter the list of simulation objects to those that can be controlled by VR-Engage players, use the VR-Engage filter (Figure 94). You can also enter VR-Engage in the Search box.

Figure 94. VR-Engage filter in Simulation Objects palette



6. Add entities, tasks, and plans as described in the *VR-Forces Users Guide*.
7. Save your scenario.

15.3. Running Your Scenario

When hosting a scenario, you can host it in either VR-Forces or VR-Engage. Use the application that best suits your needs.

VR-Engage performs better when it is hosting only the ownship in a scenario. If you have multiple users connecting different instances of VR-Engage to a scenario hosted in VR-Engage, or even if you have one user running a large scenario, performance and frame rate can deteriorate. Some entities, such as Vortex-based ground vehicles, require a minimum frame rate (approximately 20 FPS) to remain stable. You should use VR-Engage to host only simpler scenarios or for basic testing.

VR-Forces is the preferred choice when hosting a session where there will be multiple players or there are a number of entities in the scenario. Depending on the complexity of your scenario, you may want to host the simulation on one or more dedicated simulation engines. You can host these engines on one or more machines as needed to distribute the workload.

Regardless of where the scenario is hosted, VR-Engage instances can join the scenario and the players can choose a role for a controllable entity; see "3.2.1 Choosing an Existing Role" on page 53. If the scenario is hosted by VR-Forces, you can also assign roles to players as described in "3.2.4 Assigning Roles from VR-Forces" on page 56.

15.3.1 Hosting a Scenario in VR-Forces

When you host a scenario using one or more instances of VR-Forces, you would load the scenario and then have the VR-Engage stations connect to it. This works only when the connection information is the same in all instances of VR-Forces and VR-Engage.

For more information, see "3.3. Connecting to VR-Forces" on page 58



While VR-Engage can join any entity-level scenario, players can only participate in scenarios that include VR-Engage controllable entities.

15.3.2 Hosting a Scenario in VR-Engage

If you want to host your scenario in VR-Engage, such as a simple scenario for one player or for testing purposes, you simply need to copy your scenario to the *./userData/scenarios* directory. When you launch VR-Engage and choose to Host VR-Forces Session, your scenario is included in the list of available scenarios.

For more information, see "3.1.1 Hosting a VR-Forces Session" on page 49.



16. Customizing Overlays

This chapter provides examples of how you can customize the overlay for an entity.

16.1. Overview of Custom Overlays 302
16.1.1 Types of Overlays 302
16.1.2 Examples of Each Type of Overlay 302
16.2. Adding a QML Overlay 303
16.2.1 Testing Your QML Overlay 304
16.3. Adding a Custom Sensor Overlay 305
16.4. Adding a GL Studio Overlay 305
16.4.1 Creating a GL Studio RSO 306
16.4.2 Creating VR-Vantage GL Studio Updaters 306
16.4.3 Configuring the VR-Vantage GL Studio Model Definition 306
16.4.4 Modifying Role Configuration Scripts 307

16.1. Overview of Custom Overlays

There are several ways to customize the overlays for an entity's role. Depending on the complexity of the overlay, you may need to modify a configuration script, write code, or use GL Studio.

This chapter provides typical examples of customizing overlays using data that are stored as player station state attributes. It does not cover overlays that involve extensive development, or overlays that require additional data that is not currently available out of the box. If you need a complex overlay that requires C++ development, such as those used in 3D cockpits or those used with VR, AR, or MR displays, please visit https://jira.mak.com/servicedesk/customer/portal/1 and request assistance.

16.1.1 Types of Overlays

The types of overlays covered in this chapter are:

- QML overlays, which use QtQuick to create an overlay that is then referenced in the role configuration scripts. You can create complex and dynamic 2D overlays with relative ease and no additional tools. It does require some familiarity with JavaScript, the Qt Modeling Language (QML), and Lua script. Please see https://doc.qt.io/qt-5/qmlreference.html and https://www.lua.org/docs.html.
- Sensor overlays, which use the built-in capabilities of VR-Vantage and are further customized by allowing you to declare which elements you want to make visible. This also requires you to edit the Lua configuration files.
- GL Studio overlays, which are dynamic overlays that require some C++ development. This requires using the GL Studio editor from DiSTI to create an RSO. To use your own RSO, you must have a GL Studio runtime license from DiSTI. Please visit https://jira.mak.com/servicedesk/customer/portal/1 for assistance with licenses.

16.1.2 Examples of Each Type of Overlay

QML and GL Studio overlays can be dynamic and include:

- Text on the screen to reflect the state attribute of the entity.
- Rich graphics.
- User interaction (GL Studio only).

Sensor overlays allow you to have a dynamic UAV-style overlay that display:

- Target information.
- Location data.
- Reticle or cross-hairs.
- Heading, pitch, or roll information.
- Compass elements.

Detailed examples are covered in the following topics.

16.2. Adding a QML Overlay

You can add an overlay using the `DtQtQuickOverlay` component in the role definition. This component defines the QML (Qt Modeling Language) file to load and the connections between player station state attributes that are fed into QML for display in the overlay and their corresponding QML data structure names.

Player station state attributes are specified in the DtQtQuickOverlay component in the role file. You can find the player station attributes that are currently available to display in existing role files. The available attributes vary by role. Below is an example of the basic driver role using the `DtQtQuickOverlay`, found in the *./data/simulationModelSets/VR-Engage/roles/driver.lua* definition file:

```
  components = {
    ["hudMapper"] = {
      componentType = "DtGlStudioHudMapperLogic";
      priority = 10;
      modelDefinitions = {"off"};
    };
    ["quickOverlay"] = {
      componentType = "DtQtQuickOverlay";
      priority = 10;
      qmlFilename = "HUDS/groundVehicle/dashboard.qml";
      qmlDataItemObjectName = "valueSource";
      qmlDataItemBindings = {
        { stateAttribute = "speed" };
        { stateAttribute = "rpm" };
        { stateAttribute = "set-gear"; qmlProperty = "gear" };
        { stateAttribute = "resource:movement|fuel"; qmlProperty =
"fuel" };
      };
    };
 ...
  }
```

Figure 95. Car driver overlay



First, we have a hudMapper component to disable any GL Studio HUD that is currently mapped to the vehicle, `DtGlStudioHudMapperLogic` with the line `modelDefinitions = {"off"};`.

Second, we have quickOverlay component, `DtQtQuickOverlay`, to specify the QML overlay. The parameters are:

- qmlFilename. The path to the QML overlay file.
- qmlDataItemObjectName. Defines the QML item where the properties are set. This requires a corresponding `Item` in QML identified by a matching `objectName`. You can see an example of the car's `Item` in *./data/UI/HUDS/groundVehicle/dashboard.qml*.
- qmlDataBindingItems. Maps the state attributes, set in C++, to the QML item properties. The stateAttribute provides the value to use and the qmlProperty is the value to set. In the example above, the "set-gear" stateAttribute maps to the "gear" qmlProperty in the QML valueSource `Item`.



You can find the QML documentation at https://doc.qt.io/qt-5/qmlreference.html.

16.2.1 Testing Your QML Overlay

You can quickly test your QML file using the small QML test application, *qmlTestApp.exe*, which is included with VR-Engage. This app quickly loads and visualizes your QML file. To use it, navigate to *./bin64/qmlTestApp.exe* and run it with the relative path to your QML file. For example:

```
qmlTestApp.exe -d ../data/UI/HUDS/groundVehicle -f gunsight.qml
```

You can press F5 at any time to reload the file. This allows you to iteratively check whether your edits give you the look that you want.

16.3. Adding a Custom Sensor Overlay

If you need an overlay that is similar to the VR-Vantage sensor overlay (Figure 96), you can configure it by adding a `DtVrvSensorOverlayUpdater` component to a sensor operator's role definition Lua script and indicating which sections of the built-in overlay should be seen in your version.

Figure 96. Sensor overlay



For details on customizing the overlay for sensor operator roles, see "9.4. Configuring the Sensor Operator Overlay" on page 181.

> Sensor overlays pull data from the current observer. If you need sensor overlay sections in addition to those in the built-in overlay, you must use the VR-Vantage API to build a plug-in and add the section.

16.4. Adding a GL Studio Overlay

VR-Engage supports overlays created with DiSTI's GL Studio. VR-Engage uses VR-Vantage's GL Studio integration to configure and update any loaded RSOs. You must have a GL Studio license to develop a GL Studio RSO.

To add a new GL Studio overlay, you need to complete a series of procedures:

1. Create the new RSO using DiSTI's GL Studio editor.
2. Create VR-Vantage GL Studio updaters as needed.
3. Create a new VR-Vantage GL Studio model definition.
4. Modify the role configuration scripts to use the new model definition.

Please refer to DiSTI's documentation on GL Studio for details on creating an RSO.

Figure 97. GL Studio overlay



16.4.1 Creating a GL Studio RSO

To create a new GL Studio RSO, use DiSTI's GL Studio application. You can create the overlay and its properties. The properties provide a way to update values seen on the overlay or drive the dynamic components in the overlay.

The output from GL Studio is source code that is compiled into a DLL file. Save the DLL file in *./data/huds64*.

16.4.2 Creating VR-Vantage GL Studio Updaters

To update the RSO with data, you must configure the RSO's model definition with the list of VR-Vantage GL Studio updaters to provide data to the RSO.

For a list of available updaters, please see the section on cockpits in the *Creating and Editing Model Definitions* chapter in the *Adding Content to MAK Applications Reference Manual*.

16.4.3 Configuring the VR-Vantage GL Studio Model Definition

VR-Engage includes the Model Definition Editor in the Visual Model Editors dialog, which allows you to configure model definitions for GL Studio RSOs.

For more information, please see the *Model Definitions* chapter in the *Adding Content to MAK Applications Reference Manual*.

16.4.4 Modifying Role Configuration Scripts

VR-Engage entity roles are defined with Lua scripts in *./data/simulationModelSets/VR-Engage/roles*. To use your new GL Studio model definition when a role is activated, you need to add a `DtGlStudioHudMapperLogic` component to the role definition. In this example, we are using our `vreHUDPlusMFDs` model:

```
{ componentType = "DtGlStudioHudMapperLogic";
       priority = 10;
       modelDefinitions = {"vreHUDPlusMFDs"};
}
```

This component overrides the entity type mapping of VR-Vantage with the declared mapping.



17. Activating Scripts in VR-Forces

Characters in VR-Engage can activate scripts in the VR-Forces simulation engine.

17.1. Activating Lua Scripts in VR-Forces 310
17.2. Adding a Script to the Action Menu 310
17.3. Activating a Script 311

17.1. Activating Lua Scripts in VR-Forces

VR-Forces lets users add tasks and set data requests by writing Lua scripts. Lua is a scripting language that does not require compilers and software development tools like those required by the VR-Forces C++ toolkit. Also, you do not need a VR-Forces developers license to add Lua scripts.

VR-Engage characters can trigger scripts or activate them in the VR-Forces simulation engine. Once a script is triggered, there is no further involvement of the character you are playing. The script must have previously been written in VR-Forces. Then it must be added to the Action menu to be accessible to characters.

For information about how to write a script, please see the *VR-Forces Users Guide*. This chapter explains how to add a script to the Action menu and how to execute it.

17.2. Adding a Script to the Action Menu

The Action menu is configured in the file *./data/simulationModelSets/VR-Engage/roles/baseRole.lua*.

The Action menu and the Scripts submenu are configured in the following lines of the file (comments have been removed):

```
updaters={
    DtActionMenuUpdater = {
        menus={
            ["Action"] = {
                parent="";
                size={x = 400, y = 300};
                position={ x= 100, y = 300};
                elements={
                    {label = "Flashlight:", action =
"toggleFlashlight", decorator="$STATE", stateAttribute="flashlight",
stateList={"OFF", "ON"} };
                    {label = "Night Vision:", action =
"toggleNvg",    decorator="$STATE", stateAttribute="NVG", stateList=
{"OFF",    "ON"} };
                    {label = "Radio", action = "cycleRadio",
decorator="$STATE", stateAttribute="currentRadio"};
                    {label = "Task entity", action = "selectMenu",
param = "Select Entity"};
{label = "Execute Script", action = "selectMenu", param = "Scripts"};
                    ...
                };
            };
            ...
            ["Scripts"] ={
parent="Action";
    elements={
```

```
{label = "Script 1", action="scriptedTask",    param="Fire_For_Effect_
on_Location", cursor="selectLocationCursor"};
                };
            };
...
```



The `label = "Execute Script"` line might be commented out. You need to uncomment it for the Execute Script menu command to be enabled.

The Scripts menu option is added as an element of the Action menu. It has an element for each script to be listed on the submenu. In this case, there is one script with the following parameters:

- **label**. The text shown on the menu.
- **action**. Must be scriptedTask.
- **param**. The name of the Lua script.
- **cursor**. A location selected by the player. This is the only input to the script.

The only input that a VR-Engage character can provide to a script is a cursor position.

17.3. Activating a Script

You can tell VR-Forces to activate a script.

➤ **To activate a script:**

1. Open the Action menu.
2. Select Execute Script.
3. Press the **Right arrow** key, or the right button on the D-pad to show the submenu. The Scripts menu opens.
4. Select the script that you want to activate.
5. Press the **Right arrow** key or right button on the D-pad.



A. Troubleshooting

This appendix describes solutions to common issues. If your problem is not covered here, visit MAK's Support Portal to request assistance.

A.1. Performance and Display Issues 314
A.2. The Fault Tolerant Heap 314
A.3. Issues with VRAM 314
A.4. Remote Camera Placement 315
A.5. Other Limitations 315

A.1. Performance and Display Issues

There are a few issues that require you to add the application name to the *BlackApps.dat* file.

- Nahimic can severely impact performance and frame rates because it interacts with certain hardware capabilities, vantage leverages, in an extremely detrimental way.
- Under certain circumstances, opening VR-Engage on a second display prevents the program from starting.
- In addition, launching from the main display and then moving the mouse cursor to the other monitor causes the program to freeze and not respond.
- Nahimic can sometimes cause VR-Engage to crash.
- You might also experience crashes when trying to engage rotary- or fixed-wing aircraft.

If you experience any of these issues, edit the Nahimic *BlackApps.dat* file and add the application name (for example, vrengage.exe) to the list. You can search *C:\ProgramData* for *BlackApps.dat* to locate it. This file might be found in *C:\ProgramData\A-Volute\A-Volute.Nahimic\Modules\Scheduled\Configurator*, *C:\ProgramData\A-Volute\AVolute.28054DF1F58B4\Modules\ScheduledModules\Configurator*, or something similar.

A.2. The Fault Tolerant Heap

The Fault Tolerant Heap (FTH) is a subsystem on Windows that can affect software developers and testers. If a software application crashes many times—for example, in the process of programming and debugging that application—Windows adds it to the FTH. This is significantly slower than the normal heap and can have a negative impact on the performance of the application as a result.

For more information and to learn possible solutions, see https://docs.microsoft.com/en-us/windows/win32/win7appqual/fault-tolerant-heap.

A.3. Issues with VRAM

You can experience issues with hangs or crashes that are caused by high VRAM. If that is the case, use the `--limitEntityTextures` or `--limitTerrainTextures` command-line options to use only mipmap levels at or below the resolution for entities or terrain. Use two integers for the width and height. For example:

```
  vrEngage.exe --limitEntityTextures "1024 1024" --limitTerrainTextures
"1024 1024"
```

If you continue to experience crashes, use `--limitGpuMemoryPercentage` in addition to these command-line options. Typical values are between 85 and 100, for example: `--limitGpuMemoryPercentage 85`. VR-Engage does not make new textures or buffers as a percentage of the available video RAM beyond the specified percentage. This

applies only to VR-Engage-side textures: terrain insets, entities, indirected osgEarth layers (feature_geom and buildings). It does not apply to SilverLining, Triton, or osgEarth tiles. The latter are controlled by the *.earth* file options.

A.4. Remote Camera Placement

It is sometimes difficult to precisely position a remote camera on an entity. The camera may jump to a nearby spot, rather than attaching to the desired location on the entity. If this starts happening, the workaround is to place a different entity (human or other entity) in an arbitrary location on the terrain. Then return to the process of creating a new remote camera on the desired entity.

A.5. Other Limitations

There are some other limitations when using VR-Engage:

- SensorFX does not work with virtual reality devices.
- If you task the fire_team unit to perform a Move to Location task, the unit spreads out into a widely spaced formation. Doing this in a constricted area such as the warehouse area in Hawaii leads to unexpected results, as the entities struggle to get into formation.



B. Embarkation Slot Configuration in VR-Engage

VR-Engage supports the entity-level embarkation slot configuration behavior. This appendix describes how embarkation slots are configured for player roles.

B.1. Overview of Embarkation Slots 318
B.2. Door Point-type Slots for Scripted Doors on Walkable Vehicles 318
B.2.1 Ingress vs. Egress Door Points 318
B.2.2 Example Door Point Slot Configuration 318
B.2.3 Door Point Parameters 319
B.3. Embark Point-type Slots 320
B.3.1 Example Embark Point Slot Configuration 320
B.3.2 Embark Point Parameters 321

B.1. Overview of Embarkation Slots

VR-Engage is designed to be compatible with EntityLevel SMS Embarkation Slot configuration and CGF embarkation behavior. See the chapter on Configuring Embarkation in the *VR-Forces Users Guide* for instructions on configuring Embarkation Slots. In some cases, no additional Embarkation Slot configuration may be required.

When a VR-Engage player approaches an unoccupied slot position, VR-Engage presents the player with a menu showing the slot name or the name of the role associated with that slot, if there is one. If the player selects an option from the menu, they are embarked in the chosen slot and automatically engaged in the associated role. If the slot has been configured with a 0,0,0 relative position, then the slot is presented in the menu as long as the player is in proximityto the vehicle.

If this behavior is sufficient, no further slot configuration is required. However, VR-Engage vehicles can be configured with two additional special slot types, "DoorPoint" and "EmbarkPoint", for greater embarkation control.

B.2. Door Point-type Slots for Scripted Doors on Walkable Vehicles

For higher fidelity vehicles with articulated doors and walkable interior geometry, you might want to configure a special Door Point slot to act as a menu action point, giving players the ability to open or close a vehicle door. See the chapter *The WRM Specification (DIS Notes)* in the *Adding Content to MAK Applications Reference Manual* for instructions on setting up vehicles with walkable interior geometry and functional doors. A script to toggle the door is also required].

A Door Point is defined by an Embarkation Slot with the special type "DoorPoint" and an Ingress Technique, Egress Technique, or both, which corresponds to the name of the script that toggles a given door. The Door Point slot should be positioned where the player is expected to open the door from, typically at the location of the door itself. When the player approaches this point, a menu option is presented to control the door. When the player selects this option, the associated script is called to open or close the door.

B.2.1 Ingress vs. Egress Door Points

You can define a Door Point for use from inside or outside of the vehicle. If the script name appears in the Ingress Technique field, the Door Point is presented on the menu when the player is outside the vehicle. If you use the Egress Technique field, the Door Point is presented when the user is embarked on the vehicle. Defining the script name in both Ingress and Egress fields in the same Door Point slot makes that Door Point usable for both cases.

B.2.2 Example Door Point Slot Configuration

In the example, the entries in lines 2, 3, and 7 are inherited from the standard slot configuration, but are ignored on a VR-Engage `DoorPoint` type slot.

Note that this example uses both `ingressTechnique` and `egressTechnique`. This makes this Door Point functional for players both inside and outside of the vehicle.

```
 <embarkationSlot embarkableObject="-1:3:1:-1:-1:-1:-1:-1"
capacity="0"
  jumpToLocation="True" turnToHeading="True" orientOnIngress="False"
invisibleWhenEmbarked="False" embarkedAppearance=" 0XA0000"
embarkedAnimation="stand_ready">
  <embarkedOrientation yaw="0" pitch="0" roll="0"/>
  <embarkedPosition x="-4.2" y="0" z="0"/>
  <string paramName="slotName">Rear Door</string>
  <string paramName="slotType">DoorPoint</string>
  <string paramName="stationName">16</string>
  <string paramName="ingressTechnique">Toggle_Cargo_Ramp</string>
  <string paramName="egressTechnique">Toggle_Cargo_Ramp</string>
 </embarkationSlot>
```

B.2.3 Door Point Parameters

Table 32. Parameters for Door Points

Type	Name	Required	Description
Attribute	capacity	Yes	This field should always be set to 0. This prevents this special slot from being presented as a regular embarkation slot in the VR-Forces GUI embarkation dialog.
Attribute	embarkableObject	Yes	This field functions exactly as it does for any VR-Forces slot. The Door Point is only available for entities which match the type filter.
Element	embarkedPosition	Yes	This field defines the position of the Door Point on the vehicle, which controls where the player will be presented with the menu option.
Element Text	egressTechnique	No	Defines the name of the script used to control the door. When egressTechnique field is not blank, this Door Point will be presented to players approaching the door from inside the vehicle. Ingress and egressTechnique may be used together, or separately.

Table 32. Parameters for Door Points (continued)

Type	Name	Required	Description
Element Text	ingressTechnique	No	Defines the name of the script used to control the door. When ingressTechnique field is not blank, this Door Point will be presented to players approaching the door from outside of the vehicle.
Element Text	slotName	Yes	The slotName field on a Door Point slot defines the text which will appear on the menu when the player approaches the Door Point position.
Element Text	slotType	Yes	This field must be "DoorPoint". This is what defines this slot as a special VR-Engage Door Point slot.

B.3. Embark Point-type Slots

The second special slot type that VR-Engage supports is the Embark Point slot. Embark Points are used to give greater control over the position on a vehicle where the menu option to embark in a given slot is presented to the player. For embarkation slots with no associated Embark Point, the position of the embarkation slot itself is used. In many cases, this may be sufficient. A car door, for example, is near enough to the seat associated with it that a special Embark Point is probably not necessary.

On the other hand, the driver's seat in a main battle tank may not be near enough to the perimeter of the vehicle to be accessible in this manner. In that case, you can create an Embark Point and place it outside of the tank at the position where a player might expect to climb onto the tank to gain access to the driver's hatch. Embark Points are associated with slots by slotName, and multiple Embark Points can be associated with the same slot. This may be used, for example, to give a driver access from points on either side of a vehicle.

A position of 0,0,0 works for Embark Points just as it does for all embarkation slots, providing the menu option to embark in the associated slot whenever the player is in proximity to the vehicle.

B.3.1 Example Embark Point Slot Configuration

In the example, the entries in lines 2, 3, 5, 7, and 9 are inherited from the standard slot configuration, but are ignored on a VR-Engage EmbarkPoint type slot.

Unlike a Door Point, the `slotName` of the Embark Point is not used. Instead, the menu for this point shows the name of the associated embarkation slo, or the name of the role associated with that slot.

```
1   <embarkationSlot embarkableObject="-1:3:1:-1:-1:-1:-1:-1"
  capacity="0"
```

```
    jumpToLocation="True" turnToHeading="True"
orientOnIngress="False" invisibleWhenEmbarked="False"
embarkedAppearance=" 0XA0000" embarkedAnimation="stand_ready">
    <embarkedOrientation yaw="0" pitch="0" roll="0"/>
    <embarkedPosition x="0" y="0" z="0"/>
    <string paramName="slotName">Gunner</string>
    <string paramName="slotType">EmbarkPoint</string>
    <string paramName="stationName">16</string>
    <string paramName="ingressTechnique">Embark:Gunner</string>
    <string paramName="egressTechnique"/>
  </embarkationSlot>
```

B.3.2 Embark Point Parameters

Table 33. Embark Point parameters

Type	Name	Required	Description
Attribute	capacity	Yes	This field should always be set to 0. This prevents this special slot from being presented as a regular embarkation slot in the VR-Forces GUI embarkation dialog.
Attribute	embarkableObject	Yes	This field functions exactly as it does for any VR-Forces slot. That is, the Embark Point is only available for entities which match the type filter. In most cases, this field should be set to match the associated embarkation slot. If the player entity type does not match the filter on both the Embark Point and the associated embarkation slot, the Embark Point will not work.
Element	embarkedPosition	Yes	This field defines the position of the Embark Point on the vehicle, which controls where the player will be presented with the menu option to embark. A value of 0,0,0 means that the menu option will be available anywhere in proximity to the vehicle.

Table 33. Embark Point parameters (continued)

Type	Name	Required	Description
Element Text	ingressTechnique	Yes	The ingressTechnique string associates the Embark Point with an embarkation slot. This string should be in the form of "Embark:*slotName*", where *slotName* matches the slotName parameter of a regular embarkation slot.
Element Text	slotType	Yes	This field must be "EmbarkPoint". This is what defines this slot as a special VR-Engage Embark Point slot.



Index

-

-- command-line option, 73
--airMode command-line option, 73
--antiAliasing command-line option, 73
--appCacheDir command-line option, 73
--appDataDir command-line option, 73
--appIdRange command-line option, 84
--appMode command-line option, 73
--appNumber command-line option, 82, 84
--autoConnect command-line option, 74
--autoReconnectPeriod command-line option, 73
--bindObservers command-line option, 74
--dataDir command-line option, 74
--defaultObjectTimeout command-line option, 84
--depthBuffer command-line option, 74
--destAddrString command-line option, 84
--deviceAddress command-line option, 84
--disableDynamicTerrain command-line option, 74
--disableErrorDialog command-line option, 75
--disableIndirectEntities command-line option, 75
--disableKDTrees command-line option, 77
--disableOcclusionCulling command-line option, 75
--disableReverseZBuffer command-line option, 75
--disableTerrainObserverViews command-line option, 75
--discoveredModelPreloadFileName command-line option, 75
--displayEnginePort command-line option, 75
--disPort command-line option, 84
--dispSetting command-line option, 75
--disVersion command-line option, 84
--enableMultipleBackends command-line option, 72, 76
--enableSensorFX command-line option, 76
--enableTracyProfiling command-line option, 76
--enableVSync command-line option, 76
--entDispSetting command-line option, 76
--entityDefsDir command-line option, 76
--entityGridCellSize command-line option, 76
--entTypeMap command-line option, 75
--envSetting command-line option, 76
--execname command-line option, 84
--exerciseId command-line option, 85
--extraArgs command-line option, 75
--factoryRootDir command-line option, 76
--federateName command-line option, 83
--federateType command-line option, 83
--fedFileName command-line option, 82
--fileTransporterReceivePort command-line option (DIS), 83-84
--fileTransporterReceivePort command-line option (HLA), 82
--fomMapperInitData command-line option, 82
--fomMapperLib command-line option, 82
--fomModules command-line option, 82
--forceTextShaping command-line option, 76
--frameRate command-line option, 76
--fullScreen command-line option, 76
--groundMode command-line option, 77
--guiWindowRectangle command-line option, 77
--hasNTPSync command-line option, 81
--help command-line option, 77
--highestPriority command-line option, 77
--hostAddressString command-line option, 82, 84
--ignore_rest command-line option, 73
--ignoreZoomForSpeedtreeLod command-line option, 82

--initSettingsFile command-line option, 77
--limitEntityTextures command-line option, 77
--limitGpuMemoryPercentage command-line option, 78
--limitTerrainTextures command-line option, 77
--loadObserverViews command-line option, 77
--locale command-line option, 76
--localSettingsDesignator command-line option, 83
--logFileName command-line option, 77
--mainScreenNum command-line option, 78
--mcastTtl command-line option, 84
--mergeEntityTypeMap command-line option, 78
--mergeHierarchyEntityDef command-line option, 78
--mergeHierarchyTacticalGraphicsDef command-line option, 78
--mergeHierarchyUnitDef command-line option, 79
--mergeLeafEntityDef command-line option, 79
--mergeLeafTacticalGraphicsDef command-line option, 79
--mergeLeafUnitDef command-line option, 79
--mergeModelDefs command-line option, 79
--mergeTacticalGraphicsTypeMap command-line option, 79
--mergeUnitTypeMap command-line option, 79
--modelPreloadFileName command-line option, 79
--multicastAddresses command-line option, 84
--noAppDataEntityDefs command-line option, 80
--noAppDataEntityTypeMap command-line option, 80-81
--noAppDataModelDefs command-line option, 80
--noAppDataTacticalGraphicsDefs command-line option, 80
--noAppDataTacticalGraphicsTypeMap command-line option, 80
--noAppDataUnitDefs command-line option, 80
--noGui command-line option, 81
--notifyLevel command-line option, 79
--plugin command-line option, 81
--pseudoFullScreen command-line option, 81
--realizeMinimap command-line option, 81
--recvBufferSize command-line option, 84
--reportedHostAddressString command-line option, 84
--reportedHostAddressString command-line option (DIS), 83
--rprFomRevision command-line option, 83
--rprFomVersion command-line option, 83
--sendBufferSize command-line option, 85
--sensorMode command-line option, 81
--sessionId command-line option, 82
--sharedDataDir command-line option, 81
--sharedDataRoot command-line option, 81
--sharedDataStem command-line option, 81
--showConsole command-line option, 74
--siteId command-line option, 83, 85
--slave command-line option, 81
--stationName command-line option, 81
--stencilBits command-line option, 81
--subnetMask command-line option, 85
--suppressHatIntersectOnAttach command-line option, 81
--tcpSendTimeout command-line option, 82
--unplug command-line option, 82
--useAbsoluteTimeStamps command-line option, 83
--useAbsoluteTimeStamps command-line option (DIS), 85
--useAdvisories command-line option, 83
--useAsyncIO command-line option, 85
--useGeographicDdm command-line option, 84
--useIpv6 command-line option, 85
--userDataDir command-line option, 82
--version command-line option, 82
--withDisplayEngine command-line option, 82
-a command-line option, 82, 84
-A command-line option, 84
-B command-line option, 74
-C command-line option, 74
-e command-line option, 75
-E command-line option, 75
-f command-line option, 82
-F command-line option, 76
-G command-line option, 76
-h command-line option, 77
-H command-line option, 82, 84
-i command-line option, 82
-K command-line option, 77
-l command-line option, 77
-L command-line option, 77
-m command-line option, 81
-n command-line option, 79
-N command-line option, 82
-O command-line option, 81
-p command-line option, 83
-P command-line option, 84
-s command-line option, 81
-S command-line option, 81, 83
-v command-line option, 82
-W command-line option, 82
-x command-line option, 84-85
-z command-line option, 75
-Z command-line option, 82

3

3D cockpit, 151

A

accuracy
  targeting reticle, 116
Action menu, 92
  game controller mapping, 195
  mouse and keyboard mappings, 191
  opening, 92
  selecting options, 92
activating
  script, 311
add-on
  installing VR-Forces and VR-Vantage, 38
  VR-Engage for VR-Forces, 38
adding
  entities, 54
  plug-in, 86
    configuration, 88
Afghan Police Officer, 240
Afghan Soldier, 241
African Insurgent, 241
Agent M9, 241
AH-64D Apache Longbow, 240
AH-64E Apache Guardian with FCR, 240
aiming
  rifle and handgun, 116
  turret, 140
  weapon, 116
air-to-air
  combat, 158
  target
    radar, 159, 162
    sensor view, 162
air-to-ground
  combat, 157
  target
    laser, 158
    MGRS coordinate, 157
Air mode
  running in, 73
aircraft
  controlling, 152
  countermeasures, 156
  game controller mapping, 213-214
  gunner, 147
  HMD, 152
  keyboard mappings, 194
  roles, 147
altitude
  setting for autopilot, 153
Ambulance (Vortex), 242
AN/PED-1 LLDR, 242
  using, 123
anti-aliasing, 73
application
  data
    specifying location of, 73-74
  number, 82, 84
application ID, 84
application, restrictions, 73
AR, See augmented reality
assigning task, 127
attach parameter, 71
audio, 102
audio settings
  global, 103
  MAK radio, 100
augmented reality, 64, 67
  configuring, 66
  system requirements, 64
Australian Soldier EF88, 242
Australian Soldier XM107, 243
autopilot
  mode, 153
    setting, 153

B

back-end
  multiple, 76
battery
  drone, 169
Beechcraft MC-12 King Air, 243
Beechcraft MC-12 King Air Single Seater, 243
binoculars, 130
bit
  depth, 74
brake
  parking
    releasing and engaging, 138
braking
  ground vehicle, 138
Brazilian Soldier, 244
buffer
  size, 85
bullseye
  selecting, 154

C

camera
  drone view, 168
  remote, 174, 177
  Vive
    testing, 68
capturing images, 188
car
  game controller mapping, 199
carrry person, 126
CH-149 Cormorant, 244
CH-47 Chinook, 244
chaff, 156
changing
  human and drone Gamepad control, 170
  posture, 111
  role, 59
  sensor mode, 176
Chilean M16, 245
Chilean M16-M203, 245
Chilean M4, 245
Choose Role Panel, 59
choosing
  role, 52
  weapon
    aircraft, 156
    gunner, 139
    soldier, 116
Civilian Female With Drone, 246
Civilian Male With Drone, 246
climbing
  ladders, 112
cockpit
  3D, 151
Colombian Police Officer, 246
Colombian Soldier, 247
combat
  air-to-air, 158
  air-to-ground, 157
command-line option
  --, 73
  --airMode, 73
  --antiAliasing, 73
  --appCacheDir, 73
  --appDataDir, 73
  --appIdRange, 84
  --appMode, 73
  --appNumber, 82, 84
  --autoConnect, 74
  --autoReconnectPeriod, 73
  --bindObservers, 74
  --dataDir, 74
  --defaultObjectTimeout, 84
  --depthBuffer, 74
  --destAddrString, 84
  --deviceAddress, 84
  --disableDynamicTerrain, 74
  --disableErrorDialog, 75
  --disableEscapeKey, 75
  --disableIndirectEntities, 75
  --disableKDTrees, 77
  --disableOcclusionCulling, 75
  --disableReverseZBuffer, 75
  --disableTerrainObserverViews, 75
  --discoveredModelPreloadFileName, 75
  --displayEnginePort, 75
  --disPort, 84
  --dispSetting, 75
  --disVersion, 84
  --enableDofConstraints, 75
  --enableMultipleBackends, 72, 76
  --enableSensorFX, 76
  --enableTracyProfiling, 76
  --enableVSync, 76
  --entDispSetting, 76
  --entityDefsDir, 76
  --entityGridCellSize, 76
  --entTypeMap, 75
  --envSetting, 76
  --execname, 84
  --exerciseId, 85
  --extraArgs, 75
  --factoryRootDir, 76
  --federateName, 83
  --federateType, 83
  --fedFileName, 82
  --fileTransporterReceivePort (DIS), 83-84
  --fileTransporterReceivePort (HLA), 82
  --fomMapperInitData, 82
  --fomMapperLib, 82
  --fomModules, 82
  --forceTextShaping, 76
  --frameRate, 76
  --fullScreen, 76
  --groundMode, 77
  --guiWindowRectangle, 77
  --hasNTPSync, 81
  --help, 77
  --highestPriority, 77
  --hostAddressString, 82, 84
  --ignore_rest, 73
  --ignoreZoomForSpeedtreeLod, 82
  --initSettingsFile, 77
  --limitEntityTextures, 77

--limitGpuMemoryPercentage, 78
--limitTerrainTextures, 77
--loadObserverViews, 77
--locale, 76
--localSettingsDesignator, 83
--logFileName, 77
--mainScreenNum, 78
--mcastTtl, 84
--mergeEntityTypeMap, 78
--mergeHierarchyEntityDef, 78
--mergeHierarchyTacticalGraphicsDef, 78
--mergeHierarchyUnitDef, 79
--mergeLeafEntityDef, 79
--mergeLeafTacticalGraphicsDef, 79
--mergeLeafUnitDef, 79
--mergeModelDefs, 79
--mergeTacticalGraphicsTypeMap, 79
--mergeUnitTypeMap, 79
--modelPreloadFileName, 79
--multicastAddresses, 84
--noAppDataEntityDefs, 80
--noAppDataEntityTypeMap, 80-81
--noAppDataModelDefs, 80
--noAppDataTacticalGraphicsDefs, 80
--noAppDataTacticalGraphicsTypeMap, 80
--noAppDataUnitDefs, 80
--noGui, 81
--notifyLevel, 79
--plugin, 81
--pseudoFullScreen, 81
--recvBufferSize, 84
--reportedHostAddressString, 84
--reportedHostAddressString (DIS), 83
--rprFomRevision, 83
--rprFomVersion, 83
--sendBufferSize, 85
--sensorMode, 81
--sessionId, 82
--sharedDataDir, 81
--sharedDataRoot, 81
--sharedDataStem, 81
--showConsole, 74
--siteId, 83, 85
--slave, 81
--stationName, 81
--stencilBits, 81
--subnetMask, 85
--suppressHatIntersectOnAttach, 81
--tcpSendTimeout, 82
--unplug, 82
--useAbsoluteTimeStamps, 83
--useAbsoluteTimeStamps (DIS), 85
--useAdvisories, 83
--useAsyncIO, 85
--useGeographicDdm, 84
--useIpv6, 85
--userDataDir, 82
--version, 82
--withDisplayEngine, 82
-a, 82, 84
-A, 84
-B, 74
-C, 74
-e, 75
-E, 75
-f, 82
-F, 76
-G, 76
-h, 77
-H, 82, 84
-i, 82
-K, 77
-l, 77
-L, 77
-m, 81
-n, 79
-N, 82
-O, 81
-p, 83
-P, 84
-s, 81
-S, 81, 83
-v, 82
-W, 82
-x, 84-85
-z, 75
-Z, 82
command line
  loading scene, 82
  loading terrain, 82
  options, 72
commander
  LAV, 140
compass
  using, 112
configuration
  display engine
    multiple instances, 73
  entity display, 76
  environment, 76
  file
    RTI, 44
  MAK radio, 96
  plug-in
    adding, 88
    deleting, 89

world cell size, 76
configuring
display, 75
image properties, 166
joystick, 219
keyboard entity control, 219
SAR request, 165
system for MAK RTI, 44
virtual reality, 66
connecting
to network automatically, 74
controller
flight, 201
head-mounted display, 215
T.Flight Hotas 4, 207, 210
controlling
aircraft, 152
drone, 169
ground vehicle, 136
VR-Forces entity, 58
coordinateConverterOrigin parameter, 143
copilot
role, 147
countermeasures
aircraft, 156
crash, 314
creating
entity, 54
simulation model set, 278
VR-Engage scenario, 298
crouching, 111
culling
occlusion, 75
custom overlays, 302
cycle weapon
aicraft, 157
gunner, 139
soldier, 116

D

D-pad
controlling Action menu, 92
damage
indicator, 115
sustaining, 115
tracking, 114
weapon accuracy, 116
data
directory
specifying location of, 74
decibel, configuration, 104
deleting
plug-in
configuration, 89
XML file, 89
deploying
drone, 170
weapon, 119
depth
bits, 74
designate target
commander, 141
destination address, 84
detonation
event
mapping sound effects to, 105
sound effects, 102
Device address, 84
DI AK-47, 247
DI Lasing (CIS), 247
DI Lasing (US), 248
dialog box
License Setup, 41
differential
locking and unlocking, 138
directory
installed, 38
DIS
exercise ID, 85
port, 84
disabling
mouse control, 110
Disembark Entity task, 128
disembarking
vehicle, 126
display
aircraft, 151
loading configuration, 75
display engine
multiple instances, 73
display options
ground vehicle, 137
DLL, 85
plug-in
removing, 88
specifying for plug-in, 87
door
opening and closing, 133
drag person, 126
driver
ground vehicle, 137
role, 140
drone, 171
battery, 169

camera view, 168
capturing images, 188
changing Gamepad control, 170
deploying, 170
game controller mapping, 197
landing, 171
operators, 168
recovering, 171
restarting use, 170
resuming use, 170
returning home, 171
starting, 170
stopping use, 170
streaming video, 186
taking pictures, 188
using, 169
drone operator
role, 168
dynamic linked library, 85
dynamic terrain, 74

E

embarking
on vehicle, 59
Stryker, 125
vehicle, 124
emergency light, 142
enableMultipleBackends parameter, 72
enabling
laser designator, 133
mouse control, 110
sound effects, 103
ending
scenario, 61
engaging LAV gear, 137
entity
controllable
Afghan Police Officer, 240
Afghan Soldier, 241
African Insurgent, 241
Agent M9, 241
AH-64D Apache Longbow, 240
AH-64E Apache Guardian with FCR, 240
Ambulance (Vortex), 242
AN/PED-1 LLDR, 123, 242
Australian Soldier EF88, 242
Australian Soldier XM107, 243
Beechcraft MC-12 King Air, 243
Beechcraft MC-12 King Air Single Seater, 243
Brazilian Soldier, 244
CH-149 Cormorant, 244
CH-47 Chinook, 244
Chilean M16, 245
Chilean M16-M203, 245
Chilean M4, 245
Civilian Female With Drone, 246
Civilian Male With Drone, 246
Colombian Police Officer, 246
Colombian Soldier, 247
DI AK-47, 247
DI Lasing (CIS), 247
DI Lasing (US), 248
Eurofighter Typhoon, 248
F-16A Fighting Falcon, 248
F-35 Lightning II, 249
F/A-18 Hornet, 249
Farmer M9, 249
Fiat 124 (Vortex), 250
GTK Boxer APC, 250
HMMWV Utility Vehicle, 250
Insurgent, 251
Insurgent M9, 251
Iranian Soldier, 252
Iraqi Police Officer, 252
Iraqi Soldier, 252
ISIS Fighter, 251
Israeli Soldier, 253
Israeli Soldier M4, 253
Kurdish Fighter, 253
LAV III APC (Vortex), 254
Libyan Police Officer, 254
Libyan Soldier, 254
M1120A4 HEMTT LHS, 255
M1126 Stryker ICV with M2, 255
M1151 ECAC with M2, 256
M1151 ECAC with M240, 255
M1151 ECAC with M240B, 255
M1151 ECAC with MK 19, 256
M1151 Expanded Capacity Armament Carrier (ECAC), 256
M1A1 AIM (AUS), 257
M1A1 AIM (AUS) Mineplow, 256
M1A1 SA, 257
M1A2 Abrams MBT (Vortex), 257
M1A2 SEP Abrams, 258
M2A3 Bradley IFV, 258
M915 Tractor Truck, 258
M977 HEMTT Cargo Truck, 258
MH-60 Black Hawk, 259
Mideast Combatant, 260
MiG-29 Fulcrum, 259
MQ-8B Fire Scout, 259
MQ-9 Reaper UAV, 259
Pakistani Police Officer, 260
Pakistani Soldier, 260

Paratrooper, 261
Paratrooper M9, 261
Police Cruiser (Vortex), 261
Police Officer With Drone, 262
Policeman M9, 262
Remote Camera, 263
RQ-11 Raven UAV, 262
RQ-4 Global Hawk UAV, 263
RQ-7 Shadow UAV, 263
Russian Rebels, 264
ScanEagle UAV, 264
Security Guard M9, 264
Singaporean Soldier, 265
Somali Police Officer, 265
Somali Soldier, 265
Sport Hatchback Coupe, 266
Sport Standard Size Sedan, 266
Standard MidSize Sedan, 266
Sudanese Police Officer, 267
Sudanese Soldier, 267
Suspect, 267
Suspect M9, 268
Syrian Police Officer, 268
Syrian Soldier, 268
T-38C Talon, 269
Taliban, 269
Technical Truck, 269
Ugandan Police Officer, 273
Ugandan Soldier, 274
UH-60M Blackhawk, 270
US Army Javelin, 270
US Army Javelin (Tripod), 270
US Army M16, 271
US Army M16-M203, 271
US Army M4, 271
US Army M9, 272
US Border Patrol M9, 272
USMC M16, 273
USMC M16-M203, 272
USMC M4, 273
creating new, 54
creating remote camera, 177
culling, 76
display settings, 76
existing
choosing, 53
limit textures, 77
mapping sound effects to, 105
remote
placing sensor on, 178
sound effects, 102
task
assigning, 127
tracking with sensor, 175
VR-Forces
controlling, 58
entity parameter, 71
environment
loading configuration, 76
environment variable
MAK_INSTALL_DIR, 38
MAKLMGRD_LICENSE_FILE, 43
RTI_CONFIG, 44
Esc key
disable, 75
escaping
role, 60
Eurofighter Typhoon, 248
event
scenario, 102
exampleStartupConfig.lua, 70
executable
installation locations, 38
executing
script, 311
exercise ID
DIS, 85
eyepoint, See observer

F

F-16A Fighting Falcon, 248
F-35 Lightning II, 249
F/A-18 Hornet, 249
Farmer M9, 249
fatigue
experiencing, 115
stamina
indicator, 115
tracking, 114
weapon accuracy, 116
fault tolerant heap, disabling, 314
FDD file, 44
FED file, 44
name, 82
federate
name, 83
Fiat 124 (Vortex), 250
file
FED, 44
RID, 44
transporter (DIS), 83-84
transporter (HLA), 82
fire
event
mapping sound effects to, 105

sound effects, 102
firing
weapon
accuracy, 116
ground vehicle, 139
sensor operator, 177
soldier, 118
using HMD, 118
first person view mode, 101
flaps
raising and lowering, 156
flare, 156
flashlight
using, 130
FLEXlm, 40-41
flight controller, 201
Follow Me task, 128
FOM, 43
module, 82
FOM Mapper
initialization data, 82
library name, 82
frame rate, 45
setting, 76
FTH, 314
full-screen mode, 63
full screen, 81
mode, 76

G

G27 racing wheel
LAV controls, 199
G29 racing wheel
mappings, 199
game controller, See also Gamepad
generic numbering, 217
Gamepad
Action menu, 195
cars and trucks, 199
drone, 197
fixed-wing flight control, 213
generic numbering, 217
human, 196
rotary-wing flight control, 214
sensor operator, 214
vehicle, 197
gear
LAV
engaging, 137
geocentric terrain
configuring Vortex origin, 142
gesturing, 111
GL Studio
custom overlays, 305
GPU performance, 78
graphic
depth, 74
green screen, 67
grenade
throwing, 118
Ground mode
running in, 77
ground vehicle
braking, 138
controlling, 136
display options, 137
driving, 137
gunner, 139
mouse and keyboard mappings, 193
roles, 136
throttle, 138
Vortex, 276
weapon
stabilization, 140
GTK Boxer APC, 250
guiTerrain parameter, 71
gunner
aircraft, 147
commander override, 141
ground vehicle, 139
role, 139-140

H

hand controller
firing, 118
handedness, 118
mappings, 215
handgun
aiming, 116
choosing
grenade
choosing, 116
head-mounted display, See also HMD
controller, 215
head-up display, 147
display options, 151
heading
setting for autopilot, 153
headlights
turning on and off, 139
health
level indicator, 115
sustaining damage, 115
tracking, 114

weapon accuracy, 116
hiding
weapon, 119
HMD
aircraft, 152
enabling and disabling, 65
eye tracking, 65
firing weapon, 118
optimizing performance, 70
setup, 67
HMMWV Utility Vehicle, 250
host address, 82, 84
hosting
VR-Forces session, 49
hostname
license server, 41
HOTAS Warthog, 201
HTC Vive, 64, 118, 215
HUD, 147, 151
custom, 302
See also overlay, 302
human
carrying, 126
dragging, 126
game controller mapping, 196
gestures, 111
lasing, 131
mouse and keyboard mappings, 192
placing in vehicle, 126
resetting, 105

I

ID
application, 84
session, 82
image
properties
configuring, 166
indicator
health, 115
posture, 111
radar, 158
stamina, 115
infrared sensor, 131
initialization
settings, 77
injury
carry or drag wounded person, 126
place person in vehicle, 126
sustaining damage, 115
installing
on Windows, 38
default directory, 38
RTI, 44
VR-Forces plug-in, 38
instrument panel, 149
display options, 151
Insurgent, 251
Insurgent M9, 251
intercom, 93
using, 141
intersection
KD trees, 77
interval
timeout, 84
IP address, 83-84
IPV6, 85
IR, 131
IR mode
sensor, 176
Iranian Soldier, 252
Iraqi Police Officer, 252
Iraqi Soldier, 252
ISIS Fighter, 251
Israeli Soldier, 253
Israeli Soldier M4, 253

J

jamming
MAK radio frequency, 101
Javelin
shoulder-mounted, 121
tripod, 121
joining
VR-Forces session, 58
joystick
configuring, 219

K

KD tree, 77
keyboard
controlling Action menu, 92
entity control
configuring, 219
mapping
Action menu, 191
aircraft, 194
mappings, 191-193
Kurdish Fighter, 253

L

ladders
  climbing, 112
landing
  drone, 171
landing gear
  raising and lowering, 155
language
  specifying, 76
laser code
  defining
    sensor operator, 176
    soldier, 132
  tracking with sensor, 175
laser designator
  AN/PED-1 LLDR, 123
  defining laser code
    sensor operator, 176
    soldier, 132
  enabling, 133, 141
lasing
  AN/PED-1 LLDR, 123
  defining laser code
    sensor operator, 176
    soldier, 132
  display
    gimbaled sensor view, 133
  targets, 130
    commander, 141
    sensor operator, 177
    soldier, 131
LAV
  G27 racing wheel controls, 199
  gear
    engaging, 137
  roles on, 59
LAV commander, 140
LAV III APC (Vortex), 254
library
  dynamic, 85
Libyan Police Officer, 254
Libyan Soldier, 254
License Manager, 40-41
license server
  hostname, 41
License Setup dialog box, 41
light
  emergency, 142
LLDR
  lasing, 131
LLDR, AN/PED-1, 123
load balancing, 72
loading
  display engine configuration, 75
  entity display settings, 76
  environment configuration, 76
  object type mapping, 75
  plug-ins, 81, 86
  preventing, 82
  scene from command line, 82
  terrain
    from command line, 82
locale
  specifying, 76
localization
  specifying, 76
location
  bullseye
    selecting, 154
locking
  differential, 138
log file
  specifying, 77
look, 111
LSR, 176
LSS, 177
lying prone, 111

M

M1120A4 HEMTT LHS, 255
M1126 Stryker ICV with M2, 255
M1151 ECAC with M2, 256
M1151 ECAC with M240, 255
M1151 ECAC with M240B, 255
M1151 ECAC with MK 19, 256
M1151 Expanded Capacity Armament Carrier (ECAC), 256
M1A1 AIM (AUS), 257
M1A1 AIM (AUS) Mineplow, 256
M1A1 SA, 257
M1A2 Abrams MBT (Vortex), 257
M1A2 SEP Abrams, 258
M2A3 Bradley IFV, 258
M915 Tractor Truck, 258
M977 HEMTT Cargo Truck, 258
MAK radio
  configuring, 96
  intercom, 96
  jamming, 101
  VOX, 96
MAK Radio
  VR-Forces, 94

MAK RTI
  configuring, 44
MAK_INSTALL_DIR
  environment variable, 38
MAKLMGRD_LICENSE_FILE
  environment variable, 43
makRadio, 94
managing
  plug-ins, 85
mapping
  G29 racing wheel, 199
  game controller
    Action menu, 195
    drone, 197
    fixed wing, 213
    human, 196
    rotary wing, 214
    sensor operator, 214
    vehicle, 197
  keyboard, 191-193
  mouse, 191-193
  sound effects, 105
MaxBackends parameter, 72
memory, 45
menu
  Action, 92
message
  notification level, 79
MFD, 150
MGRS code
  sensor, 175
MH-60 Black Hawk, 259
Mideast Combatant, 260
MiG-29 Fulcrum, 259
minimap
  using, 113
minion parameter, 71
missile
  Javelin, shoulder-mounted, 121
  Javelin, tripod, 121
mixed reality, 64, 67
  configuring, 66
  system requirements, 64
mixed reality headset, Windows, 64
mode
  autopilot, 153
    setting, 153
  full screen, 63
  sensor
    changing, 176
  view
    view mode, 101
  windowed, 63
mode parameter, 71
monitor
  display options for aircraft, 151
motioning, 111
mouse
  controlling Action menu, 92
  mappings, 191-193
    Action menu, 191
mouse control
  enabling and disabling, 110
Move to Location task, 128
Move to My Location task, 129
moving
  on ladders, 112
  on terrain, 111
  turret, 140
  weapon accuracy, 116
MQ-8B Fire Scout, 259
MQ-9 Reaper UAV, 259
MR, 64
multi-function display, 149-150
multiple
  back-end, 76
  simulation engines, 72

N

navigation
  compass, 112
  minimap, 113
network
  connecting to
    automatically, 74
night vision goggles, 131
notification level
  specifying, 79
number key
  controlling Action menu, 92
NVG, 131
NVG mode
  sensor, 176

O

object
  count, 45
object type
  mapping
    loading, 75
observer, 293
observer views, 106
occlusion culling, 75

ocean
  synchronizing, 81
Oculus Rift, 64, 118, 215
opening
  Action menu, 92
  doors and windows, 133
operator
  drone, 168
  sensor, 174
optimizing
  performance, 45
optimizing virtual reality, 70
option
  command-line, 72
overlay
  custom, 302
    GL Studio, 305
    scripted, 303
    sensor, 305

P

Pakistani Police Officer, 260
Pakistani Soldier, 260
panel
  instrument, 149
parachuting, 127
parameter
  attach, 71
  coordinateConverterOrigin, 143
  enableMultipleBackends, 72
  entity, 71
  guiTerrain, 71
  MaxBackends, 72
  minion, 71
  mode, 71
  scenario, 71
  terrain, 71
paratrooper, 127
Paratrooper, 261
Paratrooper M9, 261
parking brake
  releasing and engaging, 138
path
  specifying for RTI, 44
paused, 61
pausing
  scenario, 60-61
performance
  anti-aliasing, 73
  fault tolerant heap, 314
  graphics
    depth, 74
  load balancing, 72
  occlusion culling, 75
  optimizing, 45, 314
  setting application priority, 77
  stencil bits, 81
pivoting tank, 142
playerStation.lua, 72
plug-in
  adding, 86
  configuration
    adding, 88
    deleting, 89
  DLL
    removing, 88
    specifying, 87
  installing VR-Forces and VR-Vantage, 38
  loading, 81, 86
  managing, 85
  prevent loading, 82
  XML file, 86
    deleting, 89
Plug-ins Editor, 87
Police Cruiser (Vortex), 261
Police Officer With Drone, 262
Policeman M9, 262
posture
  changing, 111
  indicator, 111
  weapon accuracy, 116
power
  specifying, 165
prone posture, 111
properties
  image, 166

Q

quadcopter. See drone, 169

R

radar
  Range While Scan, 159
  select multiple targets, 161
  Single Track Target, 160
  synthetic aperture, 164
  target
    air-to-air, 159, 162
  Track While Scan, 159
  warning receiver
    sounds, 154
Radar System, 158

RadarFX Server, 164
radio
  application, 94
  using, 93
  VR-Forces, 94
Range While Scan
  radar, 159
RDE, 31
receive buffer size, 84
recovering, 171
reloading, 120
remote camera
  sensor operator, 174
Remote Camera, 263
remote display engine
  connecting, 32
  multiple instances, 73
  support, 31
remote entity
  placing sensor on, 178
remote sensor
  configuring offset, 178
requesting
  SAR image, 164
resetting
  entity, 105
  sensor, 176
restarting
  drone use, 170
restoring
  entities, 105
restrict application access, 73
resuming
  drone use, 170
reticle
  accuracy indicator, 116
returning
  drone to home, 171
revision
  RPR FOM, 83
RID file, 44
rid.mtl file, 44
rifle
  aiming, 116
  choosing, 116
role
  aircraft, 147
  changing, 59
    by embarking, 59
  choosing, 52
  choosing existing, 53
  commander, 140
    gunner override, 141
    laser designator, 141
    slew to cue, 141
  copilot, 147
  creating new, 54
  driver, 140
  escaping, 60
  ground vehicle, 136
  gunner, 139-140
  on LAV, 59
  sensor operator, 174
  sling load operator, 147
RPR FOM
  revision, 83
  version, 83
RQ-11 Raven UAV, 262
RQ-4 Global Hawk UAV, 263
RQ-7 Shadow UAV, 263
RTI, 43
  definition of, 43
RTI_CONFIG environment variable, 44
Russian Rebels, 264
RWS
  radar, 159

S

SAR, 164
SAR image
  requesting, 164
SAR request
  configuring, 165
ScanEagle UAV, 264
scenario
  creating, 298
  ending, 61
  hosting
    in VR-Engage, 300
    in VR-Forces, 300
  pausing, 60-61
  state, 60
scenario event, 102
scenario parameter, 71
scene
  loading from command line, 82
screen number
  setting, 78
script
  activating, 311
  startup, 71
scripted overlays, 303
Security Guard M9, 264
selecting
  bullseye, 154

  targets, 161
send buffer size, 85
sensor
  capturing images, 188
  centering on screen, 175
  display
    gimbaled sensor view, 133
  focusing, 162, 175
  infrared, 131
  IR mode, 176
  mode
    changing, 176
  night vision, 131
  NVG mode, 176
  operator, 174
    firing weapons, 177
    lasing, 177
  placing on remote entity, 178
  remote
    configuring offset, 178
  resetting, 176
  streaming video, 186
  street
    tracking, 175
  taking pictures, 188
  tracking, 175
    resetting, 64
  tracking laser code, 175
  tracking MGRS, 175
  visual mode, 176
sensor mode
  changing
    crew member, 142
sensor operator, 174
  game controller mapping, 214
sensor overlay, 305
session
  hosting VR-Forces, 49
  ID, 82
  scenario
    creating, 298
    hosting, 300
  starting, 62
  unhosted, 50
  VR-Forces
    joining, 58
setting
  autopilot mode, 153
  frame rate, 76
settings
  initialization, 77
sight
  weapon, 116
signaling, 111
simulation
  objects, 45
simulation engine
  multiple, 72, 76
simulation model set
  new
    creating, 278
simulation network
  connecting to, 62
Simulation Object Editor, 276
Singaporean Soldier, 265
Single Track Target
  radar, 160
siren
  enabling and disabling, 142
site ID, 83, 85
slew to cue, 141
sling load operator
  role, 147
SO file, 85
Somali Police Officer, 265
Somali Soldier, 265
sound effects, 102
  decibel, 104
  enabling, 103
  mapping to entities, 105
sounds
  radar warning receiver, 154
Spark quadcopter. See drone, 169
specifying
  language, 76
  log file, 77
  notification level, 79
  power, 165
SpeedTree
  LOD
    zoom, 82
Sport Hatchback Coupe, 266
Sport Standard Size Sedan, 266
stabilizing
  turret, 140
stamina
  indicator, 115
  losing, 115
  tracking, 114
Standard MidSize Sedan, 266
standing, 111
starting
  drone use, 170
  session, 62
  VR-Engage, 49

startup
  script, 71
state
  scenario, 60
SteamVR, 64
  setup, 67
  testing camera, 68
stencil bits, 81
stopping
  drone use, 170
stowing
  weapon, 119
streaming video
  drone, 186
  receiving, 187
  sensor, 186
street
  tracking with sensor, 175
Stryker
  embarking on, 125
STT
  radar, 160
subnet mask, 85
Sudanese Police Officer, 267
Sudanese Soldier, 267
Suppress Location task, 129
Suspect, 267
Suspect M9, 268
switching
  human and drone Gamepad control, 170
synthetic aperture radar, 164
Syrian Police Officer, 268
Syrian Soldier, 268

T

T-38C Talon, 269
T.Flight Hotas 4 controller, 207, 210
T.Flight Hotas X, 201
  fixed-wing mappings, 207
  rotary-wing mappings, 210
taking pictures, 188
Taliban, 269
tank
  commander, 140
  pivoting, 142
target
  air-to-air
    radar, 159, 162
  air-to-ground
    laser, 158
    MGRS coordinate, 157
  following
    sensor view, 162
  lasing, 130
    sensor operator, 177
    soldier, 131
  select multiple, 161
task
  assigning to entity, 127
  Disembark Entity, 128
  Follow Me, 128
  Move to Location, 128
  Move to My Location, 129
  Suppress Location, 129
Technical Truck, 269
terrain
  dynamic, 74
  limit textures, 77
  loading
    from command line, 82
  moving on, 111
terrain parameter, 71
third person view mode, 101
throttle
  ground vehicle, 138
throwing
  grenade, 118
Thrustmaster Warthog
  fixed-wing mappings, 202
  rotary-wing mappings, 205
Thunderbolt, 64
time stamp, 83, 85
timeout
  interval, 84
timestamp, 83, 85
torch, 130
touch controller
  firing, 118
  handedness, 118
  mappings, 215
Track While Scan
  radar, 159
tracking
  entity, 175
  waypoint, 175
tracking sensor
  resetting, 64
triggering
  script, 311
troubleshoot, 314
truck
  game controller mapping, 199
turret
  aiming, 140

moving, 140
stabilization, 140
TWS
radar, 159

U

Ugandan Police Officer, 273
Ugandan Soldier, 274
UH-60M Blackhawk, 270
unhosted session, 50
unlocking
differential, 138
US Army Javelin, 270
US Army Javelin (Tripod), 270
US Army M16, 271
US Army M16-M203, 271
US Army M4, 271
US Army M9, 272
US Border Patrol M9, 272
user
data
specifying directory, 82
using
compass, 112
drone, 169
flashlight, 130
intercom, 141
minimap, 113
radio, 93
USMC M16, 273
USMC M16-M203, 272
USMC M4, 273

V

Valve Index, 64
Varjo, 64
vehicle
disembarking from, 126
embarking on, 59, 124
game controller mapping, 197
intercom, 141
resetting, 105
version
displaying, 82
RPR FOM, 83
video
RAM crash, 314
receiving stream, 187
video card, processor speed, 45
video stream
drone, 186
sensor, 186
view, observer, 106
viewpoint, See observer
virtual reality, 64, 67
configuring, 66
enabling and disabling, 65
optimizing, 70
system requirements, 64
tracking sensor
resetting, 64
visual mode
sensor, 176
visual sensor
vehicle crew, 142
Vive, 118
setup, 67
testing camera, 68
Vortex
configuring
for geocentric terrain, 142
ground vehicles, 276
game controller mapping, 199
vortexTerrainConfig.lua, 143
VR, 64
VR-Engage
add-on
for VR-Forces, 38
plug-in for VR-Forces, 298
starting, 49
VR-Engage Air, 73, 147
VR-Engage Ground, 77
VR-Engage Sensor Operator, 81
VR-Forces
add-ons installer, 298
create VR-Engage scenario, 298
entity
controlling, 58
radio, 94
session
hosting, 49
VR-Engage add-on
installing, 38
VR-Engage plug-in, 298
VR-Forces plug-in
installing, 38
VRAM issues, 314
VSync, 76

W

waypoint
  tracking with sensor, 175
weapon
  accuracy, 116
  aiming, 116
  aircraft
    choosing, 156
  deploying or stowing, 119
  firing
    ground vehicle, 139
    sensor operator, 177
    soldier, 118
  firing with HMD, 118
  gunner
    choosing, 139
  reloading, 120
  sight, 116
  soldier, 116
    choosing, 116
window
  opening and closing, 133
  specifying, 76
windowed mode, 63
Windows
  installing on, 38

X

Xbox controller
  Action menu, 195
  cars and trucks, 199
  drone, 197
  fixed-wing flight control, 213
  generic mappings, 217
  human, 196
  rotary-wing flight control, 214
  sensor operator, 214
  vehicle, 197
XML
  RTI configuration file, 44
XML file
  plug-in, 86
    deleting, 89
XR, See also augmented reality
XTAL, 64
  setup, 67

Z

Z-fighting, 75
zoom
  SpeedTree LOD, 82
zooming in and out, 130



VRN-2.0.1-1-220826