

VR-Forces

Users Guide

VT MÄK
A company of VT Systems



VR-Forces

Users Guide

VT MÄK
A company of VT Systems

Copyright © 2015 VT MÄK
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 VT MÄK.

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

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

All other trademarks are owned by their respective companies.

For third-party license information, please see "Third Party Licenses," on page xxii.

VT MÄK
150 Cambridge Park Drive, 3rd Floor
Cambridge, MA 02140 USA

Voice: 617-876-8085
Fax: 617-876-9208

info@mak.com

www.mak.com

Revision VRF-4.3.1-1-150630



Contents

Preface

How This Manual is Organized xv
VR-Forces Documentation xvii
MÄK Products xviii
How to Contact Us xx
Document Conventions xxi
Mouse Button Naming Conventions xxii
Third Party Licenses xxii
Boost License xxii
libXML and libICONV xxiii
Lua xxiii
Freefont OpenType Font Set xxiii
Third-Party Licenses for VR-Vantage Applications xxiii

Chapter 1. Introduction to VR-Forces

1.1. Overview 1-3
1.1.1. Entity-Level and Aggregate-Level Simulation 1-5
1.1.2. Realistic Display of Vehicles and Terrain (3D View) 1-6
1.1.3. Create Complex Scenarios 1-7
1.1.4. Entity Types Supported 1-8
1.1.5. Mission Planning 1-10
1.1.6. Entity Tasks 1-10
1.1.7. Scripted Tasks 1-11
1.1.8. Tactical Graphics 1-11
1.1.9. Terrain Agility and Composability 1-12
1.1.10. Flexible, Intuitive Graphical User Interface 1-13
1.1.11. Overlays 1-13
1.1.12. Behaviors 1-13
1.1.13. Multiple Attach Modes 1-14

1.1.14. Special Effects and Entity Information Visualization 1-14
1.1.15. Dynamic Ocean 1-14
1.1.16. Lighting Effects 1-14
1.1.17. Accurate Vehicle Positioning 1-15
1.1.18. Batch Mode Operation 1-15
1.1.19. Remote Control 1-15
1.2. The VR-Forces Toolkit 1-16
1.2.1. Plug-in Architecture 1-16
1.3. The B-HAVE Module for VR-Forces 1-17
1.4. DI-Guy 1-18
1.5. Support for External Communications Servers 1-18
1.6. Helpful Utilities 1-19
1.7. Third-Party Software and Content 1-20
1.7.1. SilverLining 1-20
1.7.2. GL Studio 1-21
1.7.3. SpeedTree 1-21
1.7.4. 3D Models, Terrain, and Graphical Content 1-22
1.7.5. OpenSceneGraph 1-22
1.7.6. osgEarth 1-23
1.8. Distributed Simulation Standards Supported 1-23

Chapter 2. Installing VR-Forces

2.1. Installing VR-Forces 2-2
2.1.1. Installing VR-Forces on Windows 2-2
2.1.2. Installing VR-Forces on a Linux System 2-3
2.1.3. Uninstalling VR-Forces on Windows 2-3
2.2. The VR-Forces Directory Structure 2-4
2.3. Installing and Setting Up the MÄK License Manager 2-5
2.3.1. Specifying the License Server 2-6
2.4. Installing an RTI 2-9
2.4.1. Installing the MÄK RTI 2-10
2.5. Localizing the Graphical User Interface 2-11
2.5.1. Translating Other Interface Files 2-14
2.5.2. Translating VR-Forces Scripts and Console Messages 2-15
2.5.3. Applying the Language Files 2-16
2.5.4. Merging Translation Files 2-16

Chapter 3. VR-Forces Simulation Concepts

3.1. The VR-Forces Program 3-3
3.2. Front-end and Back-end Concepts 3-3
3.2.1. How Front-ends and Back-ends Work Together 3-4
3.2.2. How VR-Forces Back-ends are Identified 3-5
3.2.3. VR-Forces Sessions 3-5
3.2.4. Coordinating Multiple Front-ends 3-7
3.2.5. Working with Multiple Back-ends 3-8
3.3. Scenarios 3-9

3.4. Objects 3-10
3.4.1. The Object Parameter Database 3-10
3.4.2. Local Objects and Remote Objects 3-10
3.5. Entity-Level Modeling and Aggregate-Level Modeling 3-11
3.5.1. Entity and Aggregate Terminology 3-12
3.6. Entities 3-13
3.6.1. How Entities are Identified 3-13
3.6.2. How Entities are Organized 3-15
3.6.3. How Entities Communicate 3-16
3.6.4. VR-Forces Uses a Component Architecture 3-17
3.6.5. Entity Behaviors and Tasks 3-19
3.7. Aggregate Modeling 3-20
3.7.1. Aggregates in Entity-Level Scenarios 3-20
3.7.2. Creating Higher Echelon Aggregates in Aggregate-Level Scenarios 3-24
3.8. Tactical Graphics 3-25
3.8.1. Points 3-26
3.8.2. Phase Lines 3-27
3.8.3. Routes 3-27
3.8.4. Areas 3-27
3.8.5. Obstacles 3-28
3.9. Tasks 3-28
3.10. Plans 3-29
3.11. Set Data Requests 3-30
3.12. Terrain Databases 3-30
3.12.1. Composing Terrains (and Creating MTF Files) 3-31
3.12.2. Supported File Formats 3-31
3.12.3. Feature Layers and Props 3-33
3.12.4. Coordinate Systems 3-34
3.12.5. Loading Supported Databases 3-36
3.12.6. Using Large Terrain Databases 3-36
3.13. Intervisibility (Line-of-Sight) 3-38
3.14. Representing and Managing Time in VR-Forces 3-39
3.14.1. Simulation Time 3-39
3.14.2. Time of Day 3-39
3.14.3. Exercise Clock Modes 3-40
3.14.4. Using HLA Time Management 3-41
3.15. Interactions 3-41

Chapter 4. VR-Forces Visualization Concepts

4.1. The Display Engine 4-2
4.1.1. Window Types 4-2
4.2. The Observer 4-2
4.2.1. Switching Between the 2D and 3D Projections 4-3
4.2.2. Observer Modes 4-3
4.2.3. Introduction to XR Mode 4-5

4.3. Introduction to VR-Forces Object Modeling 4-6
4.3.1. Model Formats for 3D Entities, Objects, and Effects 4-8
4.3.2. Model Sets 4-9
4.3.3. Mapping Entity Types to Element Definitions 4-9

Chapter 5. Starting VR-Forces

5.1. Starting VR-Forces 5-3
5.1.1. Starting VR-Forces from the VR-Forces Launcher 5-4
5.1.2. Starting Independent VR-Forces Executables 5-8
5.1.3. Specifying a Session ID 5-8
5.1.4. VR-Forces Startup Tutorial 5-8
5.2. The VR-Forces Window 5-11
5.2.1. Opening New Windows 5-13
5.2.2. Printing the VR-Forces Window 5-13
5.3. Managing a Front-end's Session Connection 5-14
5.3.1. Joining a Session 5-15
5.3.2. Resigning from a Session 5-15
5.3.3. Configuring Session Messages and Join at Startup 5-16
5.4. Configuring Time Management for HLA Exercises 5-17
5.4.1. RTI Requirements for Time Management 5-18
5.4.2. Enabling Time Management for the Back-end 5-18
5.5. Opening a Terrain Database 5-19
5.5.1. Loading a Terrain Database at Startup 5-21
5.5.2. Saving a Terrain Database in GDB Format 5-21
5.5.3. Closing a Terrain 5-21
5.6. Managing VR-Forces Settings 5-22
5.6.1. Synchronizing Settings Among VR-Forces Installations 5-24
5.6.2. Global Settings and Observer-Specific Settings 5-24
5.7. Exiting VR-Forces 5-24
5.8. Configuring Simulation Connections 5-25
5.8.1. Opening the Simulation Connections Configuration Dialog Box 5-25
5.8.2. Adding a Simulation Connection 5-26
5.8.3. Editing a Simulation Connection 5-27
5.8.4. Simulation Connection Parameters 5-28
5.8.5. Deleting a Simulation Connection 5-30
5.8.6. Configuring Auto Connect 5-30
5.8.7. Displaying Connection Information 5-31
5.9. Managing Plug-ins 5-32
5.9.1. Loading Plug-ins 5-32
5.9.2. Adding a Plug-in 5-35
5.9.3. Specifying the DLLs for a Plug-in 5-36
5.9.4. Adding a Plug-in Configuration 5-37
5.9.5. Deleting a Plug-in Configuration 5-37
5.9.6. Deleting a Plug-in 5-38
5.9.7. Viewing a List of Loaded Plug-ins 5-38

5.10. The VR-Forces Log Files 5-39

Chapter 6. Managing Objects

6.1. The Objects List Panel 6-2
6.2. Selecting Entities, Tactical Graphics, and Props 6-4
6.2.1. Selecting Objects in the Window 6-5
6.2.2. Selecting Objects in the Objects List Panel 6-6
6.2.3. Selecting an Entity in the Object Console Summary Panel 6-6
6.2.4. Unselecting Objects 6-7
6.3. Using Selection Groups 6-7
6.3.1. Creating a Selection Group 6-8
6.3.2. Selecting a Selection Group 6-9
6.3.3. Adding Objects to a Selection Group 6-9
6.3.4. Removing Objects from Selection Groups 6-10
6.3.5. Renaming a Selection Group 6-10
6.3.6. Importing and Exporting Selection Groups 6-10
6.3.7. Deleting a Selection Group 6-11
6.4. Displaying Tactical Graphics 6-11
6.4.1. Showing and Hiding Tactical Graphics 6-13
6.4.2. Showing and Hiding Tactical Graphics Labels 6-14
6.4.3. Displaying Height-Above-Terrain (HAT) Lines for Vertices (3D Only) 6-14
6.5. Displaying Information About an Object 6-15
6.5.1. Viewing Information for Multiple Objects 6-22
6.6. Viewing Object Counts 6-23

Chapter 7. Viewing Entities and Entity Information

7.1. Displaying Entity Labels 7-3
7.1.1. Displaying Text Labels for 3D Models 7-5
7.1.2. Turning Entity Labels On and Off 7-6
7.1.3. Pinning 3D Text Labels to the Window 7-6
7.1.4. Displaying Entity Labels for 2D Icons 7-7
7.1.5. Configuring Shadowing and Background Color for 2D Labels 7-9
7.2. Using Extended Labels 7-10
7.2.1. Adding an Extended Label 7-11
7.2.2. Changing an Extended Label's Index 7-12
7.2.3. Deleting an Extended Label 7-12
7.3. Entity Icons 7-13
7.3.1. Changing the Size of 2D Icons 7-14
7.3.2. Rotating 2D Icons to an Entity's Heading 7-15
7.4. Model Sets 7-15
7.4.1. Changing the Model Set 7-15
7.5. Exaggerating the Scale of 3D Entity Models 7-16
7.6. Hiding 3D Models 7-19
7.7. Changing Force Colors 7-19

7.8. Displaying HAT Lines for Entities (3D Only) 7-20
7.9. Displaying Threat and Sensor Range Rings 7-21
7.9.1. Enabling and Disabling the Display of Range Rings 7-23
7.9.2. Displaying Range Rings for an Entity 7-23
7.9.3. Configuring the Display of Range Rings 7-24
7.10. Displaying Entity Resources 7-26
7.10.1. Viewing Numerical Resource Data 7-26
7.11. Displaying Bounding Volumes 7-27
7.11.1. Showing Bounding Volumes Only for Selected Entities 7-28
7.12. Configuring Object Console Messages 7-28
7.12.1. Setting the Notification Level for Console Messages 7-29
7.12.2. Viewing All Console Messages 7-30
7.12.3. Answering Questions from Scripted Tasks 7-31
7.12.4. Saving Console Messages 7-31
7.12.5. Displaying an Object Console Warning Icon 7-32
7.12.6. Clearing the Object Console 7-32

Chapter 8. Displaying Entity Effects

8.1. Entity Effects in Aggregate-Level Scenarios 8-3
8.2. Using Ground Clamping 8-3
8.2.1. Configuring Ground Clamping 8-4
8.3. Trajectory Smoothing 8-5
8.3.1. Configuring Trajectory Smoothing 8-6
8.4. Inset Views 8-7
8.5. Displaying Trailing and Decal Effects (3D Only) 8-9
8.5.1. Specifying Models for Trailing Effects and Decal Effects 8-11
8.6. Displaying Track Histories 8-11
8.6.1. Configuring Track Histories 8-12
8.6.2. Displaying a Terrain Profile for a Track History 8-13
8.7. Viewing Fire and Detonation Effects 8-14
8.8. Viewing Fire and Detonation Lines 8-14
8.9. Viewing Aggregate Combat Graphics 8-15
8.10. Displaying Laser Designators 8-17
8.11. Displaying Supply Lines 8-18
8.12. Displaying Tactical Smoke 8-18
8.13. Displaying Electromagnetic Emissions 8-19
8.14. Displaying Radio Communication Lines 8-20
8.14.1. Enabling Radio Communications Graphics 8-21
8.14.2. Configuring Radio Communication Lines Lifetime and Display Mode 8-22
8.14.3. Configuring the Color of Radio Communication Lines 8-23
8.15. Displaying Cockpit Displays (3D Only) 8-25
8.16. Displaying Instrument Panels 8-27
8.16.1. Configuring Entities to use Instrument Panels 8-28
8.17. Displaying Shadows (3D Only) 8-29
8.17.1. Configuring Shadows 8-31

8.18. Using the Depth of Field Effect 8-32
8.18.1. Configuring Depth of Field 8-34
8.19. Displaying Wakes and Spray Effects 8-35
8.20. Enabling Buoyancy for Surface Entities 8-36

Chapter 9. Displaying Aggregate Entities

9.1. Selecting Aggregates 9-2
9.2. Expanding and Collapsing Aggregates 9-2
9.2.1. Expanding and Collapsing Aggregates in the Echelon View 9-4
9.2.2. Expanding and Collapsing the Echelon View by Level of Aggregation 9-4
9.3. Displaying Ghosted Icons 9-6
9.4. Displaying Aggregate Icons and Bounding Volumes 9-9
9.4.1. Specifying the Color Scheme for Aggregate Bounding Volumes 9-11

Chapter 10. The Observer and Observer Modes

10.1. Changing the Observer Mode 10-2
10.2. Creating and Editing Observer Modes 10-2
10.2.1. Creating an Observer Mode 10-3
10.2.2. Deleting an Observer Mode 10-4
10.2.3. Editing an Observer Mode 10-4
10.2.4. Renaming an Observer Mode 10-4
10.3. Adding an Observer 10-5
10.3.1. Adding an Observer on the Observer Settings Page 10-5
10.3.2. Adding an Observer in the Display Engine Configuration Editor 10-6
10.4. Removing an Observer 10-8
10.5. Communicating Information About the Observer 10-8
10.5.1. Making the Observer Visible to Other Applications 10-8
10.5.2. Viewing and Attaching to Other Observers 10-9
10.6. Displaying Models for Remote Observers 10-9

Chapter 11. Moving the Observer

11.1. Moving the Observer 11-3
11.1.1. Using the Compass to View the Observer's Heading 11-4
11.1.2. Editing the Compass's Attributes 11-5
11.2. Observer Movement Functions and Keyboard Mappings 11-6
11.2.1. 3D Observer Movement Coordinate Systems 11-6
11.2.2. 2D Observer Coordinate System 11-7
11.2.3. The Attached Context and Observer Movement (3D Only) 11-7
11.3. Moving the Observer from the Keyboard in the 3D View 11-9
11.3.1. Navigation Options 11-11
11.3.2. Orbiting 11-12
11.3.3. Moving the Camera 11-12

11.4. Moving the Observer Using the Mouse (3D View) 11-13
11.4.1. Dragging the Terrain 11-13
11.4.2. Changing the Observer's Orientation with the Mouse 11-14
11.4.3. Dragging the View 11-14
11.4.4. Orbiting the Terrain with the Mouse 11-14
11.4.5. Teleporting the Observer 11-15
11.5. Moving the Observer in the 2D View 11-15
11.5.1. Moving the Observer from the Keyboard in the 2D View 11-15
11.5.2. Moving the Observer Using the Mouse in the 2D Projection 11-16
11.6. Zooming the Observer 11-16
11.6.1. Zooming the Observer in the 3D Projection 11-17
11.6.2. Zooming the Observer in the 2D Projection 11-18
11.6.3. Setting the Scale Factor for Paged LODs 11-18
11.6.4. Zooming to Extents 11-18
11.7. Centering the Observer on an Entity 11-19
11.8. Forcing Orientation North 11-20
11.9. Enabling and Disabling View Constraints (3D Only) 11-21
11.10. Controlling Navigation Speed 11-21
11.10.1. Enabling or Disabling Speed Scaling 11-22
11.10.2. Changing the Observer's Movement Speed 11-23
11.11. Saving and Recalling Views 11-23
11.11.1. Recalling a View 11-25
11.11.2. Deleting Views 11-25
11.11.3. Specifying the Startup View 11-25
11.11.4. Saving Views to a File 11-26
11.11.5. Loading Views 11-26
11.12. Controlling the Observer from Other Applications 11-27
11.12.1. Enabling the Processing of View Control Messages 11-27

Chapter 12. Attaching the Observer to Objects

12.1. Attaching the Observer to an Entity or Prop 12-2
12.1.1. Attaching to an Articulated Part 12-4
12.1.2. Attaching the Observer in Mimic Track or Tether Track Mode 12-4
12.1.3. Filtering the Object Attachment List 12-4
12.1.4. Detaching from an Entity 12-5
12.1.5. Hiding the Attached Model (3D Projection Only) 12-5
12.1.6. Attaching to Props 12-5
12.1.7. Improving Performance when Attaching to Ground Entities 12-5
12.2. Attach Modes in the 3D View 12-6
12.2.1. Absolute Mode 12-6
12.2.2. Follow Mode 12-7
12.2.3. Mimic Mode 12-8

12.2.4. Mimic Track Mode 12-9
12.2.5. Space Follow Mode 12-10
12.2.6. Tether Mode 12-11
12.2.7. Tether Track Mode 12-12
12.2.8. Track Mode 12-13
12.3. Attach Modes in the 2D View 12-14
12.4. Attaching the Observer at Startup 12-14

Chapter 13. Using Terrain Profiles

13.1. Terrain Profiles 13-2
13.1.1. Displaying a Line's Terrain Profile 13-3
13.1.2. Configuring Terrain Profiles 13-3
13.2. Analyzing Terrain Using a Terrain Profile Line 13-5

Chapter 14. Intervisibility

14.1. Types of Intervisibility 14-2
14.1.1. Point-to-Point Intervisibility 14-2
14.1.2. Intervisibility Fan 14-3
14.1.3. Entity Intervisibility 14-3
14.2. Intervisibility Objects can be Transient or Permanent 14-4
14.2.1. Creating a Transient Intervisibility Object 14-5
14.2.2. Creating a Permanent Intervisibility Object 14-6
14.3. Editing an Intervisibility Object 14-7
14.4. Configuring Intervisibility Lines and Fans 14-8
14.5. Pinning an Intervisibility Object to an Entity 14-9
14.6. Showing and Hiding Intervisibility Lines 14-9
14.7. Displaying Intervisibility Line Information 14-10
14.8. Displaying an Intervisibility Line's Terrain Profile 14-11
14.9. Deleting Intervisibility Objects 14-11

Chapter 15. Sound Effects

15.1. Introduction to Sound Effects 15-2
15.2. Enabling Sound Effects 15-2
15.2.1. Setting the Sound Volume 15-3
15.2.2. Specifying the Range in which Sounds are Heard 15-3
15.3. Associating Sound Files with Entities and Events 15-4
15.3.1. Mapping a Sound File to an Entity 15-5
15.3.2. Mapping Sound Files to Fire and Detonation Events 15-8

Chapter 16. Using Sensors

16.1. Using Sensors 16-2
16.2. Enabling Sensors 16-3
16.3. Configuring Sensors 16-4
16.4. Adding New Sensors 16-6

Chapter 17. Lighting Effects

17.1. Displaying Lighting Effects 17-2
17.2. Displaying Dynamic Lighting 17-3
17.2.1. Enabling and Disabling Dynamic Lighting and Shading 17-4
17.2.2. Specifying the Emissive Lighting Scale 17-5
17.3. Displaying Ocean Planar Reflections 17-6
17.3.1. Configuring Ocean Planar Reflections 17-7
17.4. Lens Flare 17-8
17.5. Displaying Crepuscular (Sun) Rays 17-9
17.6. Configuring Ambient and Diffuse Lighting 17-10
17.6.1. Configuring the Ambient and Diffuse Scale Values 17-10
17.6.2. Configuring the Material Ambient and Diffuse Values 17-10
17.7. Configuring the Skybox Cube Map 17-12
17.8. Flashlight Mode 17-12
17.9. Displaying Shadows (3D Only) 17-13
17.9.1. Configuring Shadows 17-15
17.10. Disabling Lighting and Clouds 17-15

Chapter 18. Displaying Terrain Information

18.1. Introduction 18-2
18.2. Configuring Visible Surfaces (GDB Soil Types) 18-2
18.3. Coloring Terrain Based on Elevation 18-4
18.3.1. Configuring Elevation Color Settings 18-5
18.4. Displaying Grid Lines 18-7
18.4.1. Configuring Grid Lines 18-8
18.5. Displaying Contour Lines 18-9
18.5.1. Configuring Contour Lines 18-10
18.6. Displaying Models and Terrain in Wireframe Mode 18-10
18.7. Filtering Earth File Layers 18-12
18.8. Displaying Feature Data (2D) 18-13
18.8.1. Displaying a List of Vector Features 18-14
18.8.2. Changing the Colors Used for Feature Data 18-15
18.9. Displaying Coordinates and Soil Type for a Location 18-16
18.10. Viewing the Map Scale 18-17
18.11. Terrain Databases Provided with VR-Forces 18-17

Chapter 19. Remote Graphics

19.1. Displaying Remote Graphics 19-2
19.1.1. Viewing a List of Remote Graphics 19-4

Appendix A. Common GUI Controls and Behaviors

A.1. Dockable Panels and Toolbars A-2
A.1.1. Docking and Undocking a Panel A-2
A.1.2. Docking and Undocking a Toolbar A-3
A.1.3. Displaying and Hiding Panels and Toolbars A-3

A.1.4. Restoring the Layout of Toolbars and Panels A-4
A.2. Using Tear-Off Menus A-5
A.3. Viewing the Window in Full Screen Mode A-6
A.4. Using Context-Sensitive Menus A-6
A.5. Common VR-Forces Buttons A-7
A.6. Adding Text to the Title Bar A-8
A.7. Specifying Display Units A-9

Appendix B. Command-line Options

B.1. Command-Line Options for vrfGui B-3
B.2. Command-line Options for vrfSim B-7
B.3. Command-line Options for vrfLauncher B-12
B.3.1. Running in Combined Mode without the Launcher Dialog Box B-12
B.3.2. Using the -- Command-line Argument B-13
B.4. Protocol-Independent Command-line Options B-13
B.4.1. Specifying the Site ID and Application Number B-13
B.4.2. Specifying the Language to Use in the GUI B-14
B.4.3. Setting the Notification Level B-14
B.4.4. Loading Plug-ins B-15
B.5. Command-line Options for HLA Federations B-15
B.5.1. Specifying the Federation Execution B-15
B.5.2. Specifying the FED File B-16
B.5.3. Specifying the RPR FOM Version B-16
B.5.4. Specifying a FOM Mapper B-16
B.5.5. Specifying the RPR FOM Revision B-16
B.5.6. Specifying FOM Mapper Initialization Data B-17
B.5.7. Specifying a FED File That is Appropriate for the FOM Mapper B-17
B.5.8. Enabling Time Management B-17
B.6. Command-line Options for DIS Exercises B-18
B.6.1. Specifying the Port Number B-18
B.6.2. Specifying Point-to-Point or Multicast Operation B-18
B.6.3. Specifying the Multicast Time-to-Live B-18
B.6.4. Using Asynchronous I/O B-18
B.6.5. Specifying the Exercise ID B-19
B.6.6. Specifying the DIS Version B-19

MÄK Products Glossary

Index

VR-Forces Quick Reference Card



Preface

This manual is written for persons who will install VR-Forces® or use the VR-Forces application. The manual assumes that you are familiar with your operating system and that you know how to work in your computer's graphical window environment.

Please see release documentation for information about changes and updates to VR-Forces since this manual went to press.

How This Manual is Organized

The chapters are organized as follows:

Chapter 1, *Introduction to VR-Forces*, briefly describes VR-Forces and its features.

Chapter 2, *Installing VR-Forces*, explains how to install VR-Forces, set up the FLEXlm license manager, and localize the GUI.

Chapter 3, *VR-Forces Simulation Concepts*, provides the conceptual background for the simulation process. It will help put some of the discussion in this manual into context. It is even more relevant to *VR-Forces Scenario Management Guide*.

Chapter 4, *VR-Forces Visualization Concepts*, discusses the 2D and 3D interfaces and the concepts that underlie their use and configuration.

Chapter 5, *Starting VR-Forces* describes the command-line options for vrfGui (the front-end) and vrfSim (the back-end or simulation engine).

Chapter 6, *Managing Objects* describes procedures common to entities, control objects, and overlay objects.

Chapter 7, *Viewing Entities and Entity Information* explains how entities are displayed, how you can change the display of entities, and how you can display information about entities.

Chapter 8, *Displaying Entity Effects* describes how to display a variety of entity effect such as trailing effects, shadows, and sensor volumes.

Chapter 9, *Displaying Aggregate Entities* describes how to view aggregate entities.

Chapter 10, *The Observer and Observer Modes* explains what an observer is and the different ways observers can be configured.

Chapter 11, *Moving the Observer* explains how to navigate the 3D and 2D views by moving the observer.

Chapter 12, *Attaching the Observer to Objects* explains how to attach the observer to entities and objects. This is an alternative to manually moving the observer through the terrain.

Chapter 13, *Using Terrain Profiles* explains how to display and configure terrain profiles.

Chapter 14, *Intervisibility* explains how to determine if entities can see other entities or areas of the terrain.

Chapter 15, *Sound Effects* explains how to enable and configure sound effects for entities.

Chapter 16, *Using Sensors* explains how to enable and configure infrared and night vision goggle sensor views.

Chapter 17, *Lighting Effects* describes the lighting effects supported by VR-Forces and how to manage them.

Chapter 18, *Displaying Terrain Information* describes ways to change how terrain is displayed. These options provide different types of information about the terrain.

Chapter 19, *Remote Graphics* explains how to enable display of graphics generated by remote applications.

Appendix A, *Common GUI Controls and Behaviors* explains how to manage toolbars and other aspects of the VR-Forces GUI.

Appendix B, *Command-line Options*, describes how to start VR-Forces and load terrain databases, and configure HLA time management.

MÄK Products Glossary defines terms common to DIS, HLA, and MÄK products.

VR-Forces Quick Reference Card lists command line options, keyboard shortcuts, and navigation and view options.

VR-Forces Documentation

VR-Forces documentation is provided as manuals in PDF format, online help, and HTML class documentation. The PDF files are in the *./doc* directory. The VR-Forces documentation set is as follows:

- *VR-Forces Documentation Center* is your central starting point for accessing the VR-Forces manual set. It has a documentation roadmap, tables of contents for all manuals, and a master index to help you find references to subjects that are covered in two or more manuals or locate the manual in which a particular topic is discussed. All table of contents entries and index entries are live links to the manuals.
- *VR-Forces Users Guide* describes how to install VR-Forces and configure license management. It explains how to use the VR-Forces graphical user interface to view simulations and how to manage aspects of VR-Forces that are not directly related to creating and running scenarios.
- *VR-Forces Scenario Management Guide* explains how to create and run scenarios.
- *VR-Forces Configuration Guide* explains advanced features for configuring performance and how to edit VR-Forces configuration files. It includes documentation of the Entity Editor, OPD Editor, and Scenario Merge tools. It also explains how to compose terrains and configure 3D models.
- *VR-Forces Migration Guide* collates API migration information for recent releases.
- Online help. The VR-Forces front-end, the OPD Editor, and the Entity Editor have online help accessible from the Help menu.
- *VR-Forces Developers Guide* and API documentation. Class documentation and developers guides in linked HTML pages.
- *VR-Forces Release Notes*.
- VR-Forces *First Experience Guide*. A brief introduction to the most basic features of VR-Forces. Provides less detail than *VR-Forces Getting Started Guide*.
- *VR-Forces Entity Model Catalog*. A catalog of all of the entities configured in the Entity Editor with basic parameter details and a screen capture of the model.

MÄK Products

VR-Forces is a member of the VT MÄK line of software products designed to streamline the process of developing and using networked simulated environments. The VT MÄK 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.

- MÄK RTI. An RTI (Run-Time Infrastructure) is required to run applications using the High Level Architecture (HLA). The MÄK 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 a 2D and 3D views of a simulated environment.

  You can create and view local entities, aggregate them into hierarchical units, assign tasks, set state parameters, and create plans that have tasks, set statements, and conditional statements. VR-Forces also functions as a plan view display for viewing remote entities 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 three end-user applications (VR-Vantage Stealth, VR-Vantage PVD, and VR-Vantage IG) and the VR-Vantage Toolkit.
  - VR-Vantage Stealth 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 entities so that it moves as they do.
  - VR-Vantage IG is a configurable desktop image generator (IG) for out the window (OTW) scenes and remote camera views. It has most of the features of the Stealth, but is optimized for its IG function.
  - VR-Vantage PVD provides a 2D plan view display. It gives you the big picture of the simulated world.

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

- MÄK 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™. The DI-Guy product line is a set of software tools for real-time human visualization, simulation, and artificial intelligence. Every DI-Guy software offering comes with thousands of ready-to-use characters, appearances, and motions. DI-Guy enables the easy creation of crowds and individuals who are terrain aware, autonomous, and react intelligently to ongoing events. Save time, money and create outstanding simulations with DI-Guy. The DI-Guy product line includes the following products:
  - The DI-Guy SDK. Embed the DI-Guy library in your real-time application and populate your world with lifelike human characters.
  - DI-Guy Scenario™. Author and visualize human performances in a rich, user-friendly graphical environment. Use DI-Guy Scenario as an end visualization application or save scenarios and load them into your DI-Guy SDK enabled application.
  - DI-Guy AI. Generate crowds of autonomous characters to quickly populate your worlds with hundreds and thousands of terrain-aware, collision avoiding DI-Guys. Used as a module on top of DI-Guy Scenario and DI-Guy SDK.
  - Expressive Faces Module. Enable DI-Guy characters to have faces that display emotion, eyes that look in directions and blink, and lips that sync to sound files.
  - DI-Guy Motion Editor. Create or customize motions to your particular needs in an easy-to-use graphical application.

- RadarFX. RadarFX is a client-server application that can 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. VR-Forces includes a sample client application.

How to Contact Us

For VR-Forces technical support, information about upgrades, and information about other MÄK 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
VR-Vantage support:	vrv-support@mak.com

Internet

MÄK web site home page:	www.mak.com
License key requests:	www.mak.com/support/get-licenses.html
Product version and platform information:	www.mak.com/support/product-versions.html
For the free, unlicensed MÄK RTI:	www.mak.com/resources/bonus-material/cat_view/16-bonus-materials/24-mak-high-performance-rti.html
MÄK Community Forum:	www.mak.com/community-forum/1-forum.html

Post

Send postal correspondence to:	VT MÄK 150 Cambridge Park Drive, 3rd Floor Cambridge, MA, USA 02140

When requesting support, please tell us the product you are using, the version, and the platform on which you are running.

Document Conventions

This manual uses the following typographic conventions:

`Monospaced`	Indicates commands or values you enter.
`Monospaced Bold`	Indicates a key on the keyboard.
`Monospaced Italic`	Indicates command variables that you replace with appropriate values.
Blue text	A hypertext link to another location in this manual or another manual in the documentation set.
`{ }`	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 MÄK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames. If you are running on a PC, substitute a \ (backslash) when you type pathnames.
Italic	Indicates a file name, pathname, or a class name.
sans Serif	Indicates a parameter or argument.
➤	Indicates a one-step 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**.
	Click the icon to run a tutorial video in the default browser.
	Indicates supplemental or clarifying information.
	Indicates additional information that you must observe to ensure the success of a procedure or other task.
	Indicates that a section is valid only for entity-level scenarios.
	Indicates that a section is valid only for aggregate-level scenarios.

Directory names are preceded with dot and slash characters that show their position with respect to the VR-Forces home directory. For example, the directory *vrforces4.3.1/doc* appears in the text as *./doc*.

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

MÄK software products may use code from third parties. This section contains the license documentation required by these third parties.

Boost License

VR-Link, and all MÄK software that uses 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 MÄK software that uses VR-Link, links in libXML and libICONV. On some platforms the compiled libraries and header files are distributed with MÄK Products. MÄK has made no modifications to these libraries. For more information about these libraries please 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/.

Lua

Some MÄK products use the Lua programming language (www.lua.org). Its license is as follows:

Copyright © 1994–2012 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

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 AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Freefont OpenType Font Set

VR-Vantage applications and VR-Forces use the Freefont OpenType font set from the Free Software Foundation. It is covered by the General Public License (GPL). For details, please see: http://www.gnu.org/licenses/gpl.html

Third-Party Licenses for VR-Vantage Applications

VR-Vantage applications use a variety of third-party libraries. Developers who want to use these libraries may be required to purchase developer's licenses. Please see section 1.2 in *VR-Forces Front-End Developers Guide* for details.



1. Introduction to VR-Forces

This chapter provides a high-level description of VR-Forces features.

Overview 1-3
- Entity-Level and Aggregate-Level Simulation 1-5
- Realistic Display of Vehicles and Terrain (3D View) 1-6
- Create Complex Scenarios 1-7
- Entity Types Supported 1-8
- Mission Planning 1-10
- Entity Tasks 1-10
- Scripted Tasks 1-11
- Tactical Graphics 1-11
- Terrain Agility and Composability 1-12
- Flexible, Intuitive Graphical User Interface 1-13
- Overlays 1-13
- Behaviors 1-13
- Multiple Attach Modes 1-14
- Special Effects and Entity Information Visualization 1-14
- Dynamic Ocean 1-14
- Lighting Effects 1-14
- Accurate Vehicle Positioning 1-15
- Batch Mode Operation 1-15
- Remote Control 1-15

The VR-Forces Toolkit 1-16
- Plug-in Architecture 1-16

The B-HAVE Module for VR-Forces 1-17

DI-Guy 1-18

Support for External Communications Servers 1-18
Helpful Utilities 1-19
Third-Party Software and Content 1-20
SilverLining 1-20
GL Studio 1-21
SpeedTree 1-21
3D Models, Terrain, and Graphical Content 1-22
OpenSceneGraph 1-22
osgEarth 1-23
Distributed Simulation Standards Supported 1-23

1.1. Overview

VR-Forces is a computer generated forces (CGF) application and toolkit. VR-Forces is implemented using separate front-end (graphical user interface) and back-end (simulation engine) applications that allow tremendous flexibility for managing your simulation needs. (For information about how this works, please see "The VR-Forces Program," on page 3-3.)

VR-Forces provides 3D and 2D views of your simulated world, integrated into one graphical user interface (GUI). The 2D view provides you with insights into a simulated battle by overlaying entities and information onto 2D views of tactical, strategic, and visual databases. The 2D view answers typical questions about the placement of forces and how the terrain might affect the engagement more easily than do three-dimensional displays. It is ideally suited for viewing simulations over large areas of open terrain.

The 3D view provides three-dimensional, perspective views of the terrain and entity models. It is ideal for scenarios that require situational awareness, simulation debugging, or after action review. It is particularly useful for placing entities with precise location and orientation and for working in terrains in which entities can be at multiple altitudes at a given X, Y coordinate, such as urban terrains.

The 3D view includes exaggerated reality (XR) features. Model scaling and additional model sets help to provide a common operational picture of the simulated world.

VR-Forces is compatible with both the Distributed Interactive Simulation (DIS) and High Level Architecture (HLA) simulation standards. It supports multiple database formats and terrain servers. VR-Forces is interoperable with other MÄK products, such as VR-Vantage Stealth and MÄK Data Logger.

The VR-Forces front-end is built with the VR-Vantage Toolkit. The VR-Vantage Toolkit is based on OpenSceneGraph[1], so you can leverage value-added plug-ins built by the OSG community and MÄK partners.

1. "OpenSceneGraph is an Open Source, cross-platform graphics toolkit for the development of high-performance graphics applications such as flight simulators, games, virtual reality and scientific visualization. It is based on the concept of a SceneGraph, providing an object-oriented framework on top of OpenGL." (http://www.openscenegraph.org/projects/osg/wiki/About/Introduction)

Table 1-1 lists VR-Forces visual features supported in the 3D and 2D views for entity-level scenarios.

Table 1-1: VR-Forces features by projection (Sheet 1 of 2)

Feature	3D	2D
Terrain agility and composability	X	X
Attach modes	X	X
Observer movement modes		
Observer frame	X	X
Level observer frame	X	
Laptop	X	X
2D level observer frame		X
Model sets		
3D models	X	X (top view)
3D colorized models	X	X (top view)
2D icons	X	X
Other Features		
Advanced lighting	X	
Audio	X	X
Cockpit displays	X	
Communications lines	X	X
Detonation and muzzle flashes	X	
DI-Guy characters	X	
Display units configuration	X	X
Dynamic ocean	X	
Emitter volumes	X	X
Entity labels	Text panels	Configurable symbol decorations
Environment effects	X	
Fire and detonation lines	X	X
Forward+ lighting	X	
Intervisibility	X	X
Paged terrain	X	X
Range rings	X	X
Remote draw API	X	X (2D graphics only)

Table 1-1: VR-Forces features by projection (Sheet 2 of 2)

Feature	3D	2D
Sea spray effects	X	
Sensors	X	X (Stealth only)
Shader-based effects maps	X	
Shadows	X	
SpeedTrees	X	
Streaming buoys and beacons	X	X
Streaming terrain	X	X
Tactical graphics	X	X
Tactical smoke	X	
Terrain profiles	X	X
Track histories	X	X
Trailing effects	X	
View controls	X	X
Wireframe view	X	X
Zoom	X	X

1.1.1. Entity-Level and Aggregate-Level Simulation

VR-Forces supports two approaches to simulation – entity-level simulation and aggregate-level simulation. In entity-level simulations, VR-Forces models individual entities, such as ships, tanks, automobiles, airplanes, and people. Interaction, and for military simulations, warfare, takes place between individual entities. You can aggregate entities into higher order units, but for the most part they are still modeled as individuals.

Aggregate-level simulations do not model individual entity interactions and combat. They are concerned with the high level movement and interaction of large numbers of personnel and equipment. Aggregate-level combat is modeled based on the relative strengths and weaknesses of aggregate units in personnel, equipment, and location, not on the engagement of the individual entities that they represent. Aggregate-level simulations are used for modeling the operational tempo of large area/theater level missions overseen by command staff level officers. This is useful for training staff officers and stimulating Command and Control systems (for example, C2, C4I, C4ISR, and Mission Command systems). (Aggregate level simulations include some individual entity models to facilitate certain types of munition attacks. However, they use the same modeling approach as aggregate units. They do not behave like the comparable entities in entity-level simulations.)



Most of the information in VR-Forces documentation applies to both types of modeling. The procedures for creating scenarios, adding simulation objects and tactical graphics, assigning tasks and writing plans, and displaying scenario information are almost identical in both types of scenarios. Therefore, if you want to know how to create a scenario that uses aggregate-level modeling, for example, just follow the procedures in Chapter 1, *Creating and Running Scenarios*, in *VR-Forces Scenario Management Guide*. Or, if you want to learn about assigning tasks in entity-level scenarios, just read Chapter 7, *Assigning Tasks*, in *VR-Forces Scenario Management Guide*.

If a section only applies to entity-level simulations, it has an entity icon ( ) in the margin. If a section only applies to aggregate-level simulations, it has an aggregate icon () in the margin.

For conceptual information about the differences between entity-level modeling and aggregate-level modeling, please see the following sections:

- "Entity-Level Modeling and Aggregate-Level Modeling," on page 3-11.
- "Aggregate Modeling," on page 3-20.
- Chapter 5, *How Aggregate-Level Modeling Works*, in *VR-Forces Scenario Management Guide*.

For information specific to the different types of modeling, please see the following sections:

- Chapter 8, *Tasks for Entity-Level Scenarios*, in *VR-Forces Scenario Management Guide*.
- Chapter 9, *Tasks for Aggregate-Level Scenarios*, in *VR-Forces Scenario Management Guide*.
- Chapter 7, *Editing Aggregates for Entity-Level Scenarios*, in *VR-Forces Configuration Guide*.
- Chapter 8, *Editing Aggregate-Level Objects*, in *VR-Forces Configuration Guide*.

1.1.2. Realistic Display of Vehicles and Terrain (3D View)

VR-Forces provides a realistic display using terrain and moving-model geometry in the OpenFlight format. It includes many 3D vehicle models, some of which display movement of articulated parts, such as turrets and rotors. These models can change to show a damaged or destroyed state. You can provide your own models and map them to entity types. For details about terrain, please see Chapter 13, *Composing Terrains*, in *VR-Forces Configuration Guide*. For details about models, please see Chapter 16, *Mapping Models and Effects*, in *VR-Forces Configuration Guide*. (Aggregate-level scenarios do not use 3D vehicle models.)

1.1.3. Create Complex Scenarios

VR-Forces lets you create complex scenarios to meet your simulation requirements. You can:

- Create and delete entities.
- Create, delete, and edit tactical graphics.
- Assign a sequence of tasks and set data requests to an entity as a plan.
- Assign tasks and set data requests independently of plans.
- Create global plans that run independently of any entity.
- Observe the behavior of entities in response to their environment.
- Create, save, and load scenarios that consist of:
  - Terrain database information.
  - An order of battle.
  - Tactical graphics.
  - A plan for each entity in the order of battle.
- Manage hostility relationships.
- Manage spot reports.
- Determine line-of-sight between entities.
- Aggregate and disaggregate, and expand and collapse aggregate entities.
- Create overlays and add graphics to them.
- Schedule scenario events to inject non-simulated input, such as text, video, and graphics to scenario participants.
- Apply weather conditions to a particular area of the terrain.

All aspects of scenario creation are described in *VR-Forces Scenario Management Guide*.

1.1.4. Entity Types Supported

Using VR-Forces, you can interactively add individual entities to a simulation and you can aggregate them into higher echelon units. Entities provided for entity-level scenarios include:

- Ground entities, such as:
  - M1A2 Tank
  - T-80 Tank
  - HMMWV Utility Vehicle
  - BTR-80 APC
  - M2A2 Bradley IFV
  - IEDs
  - Civilian vehicles, such as automobiles, police cars, and fire engines
- Air entities, such as:
  - F-16 Falcon
  - F/A-18 Hornet
  - A-10 Thunderbolt
  - VTUAV Cypher
  - MiG-29 Fulcrum
  - AH-64 Apache
  - Civilian aircraft, such as Boeing 747, and Airbus A320
- Surface and subsurface entities, such as:
  - AAVC7A1 Landing craft
  - LCAC
  - Aircraft carrier
  - Guided missile frigate
  - Submarine
  - Sailboats
  - Cargo ships.
- Life forms, such as:
  - Dismounted infantry
  - Civilian
- Aggregate entities, such as platoons and companies.

Aggregate-level scenarios have entity types such as:

- ADA PLT
- Air Attack company
- Artillery battery
- Engineering company
- Ground attack aircraft
- Mechanized infantry company
- Mortar platoon
- Rifle company
- Tank company
- Destroyer
- Submarine.

Entity States

You can interactively modify aspects of an entity's state, including:

- Speed.
- On-board stores such as fuel and ammunition.
- Location.
- Heading.

For more information about changing an entity's state, please see Chapter 10, *Setting Entity State*, in *VR-Forces Scenario Management Guide*.

You can modify entity parameters by editing the object parameter database and other configuration files. For more information, please see *VR-Forces Configuration Guide*.

Embarkation

Entities can embark on and disembark from other entities, for example, dismounted infantry can embark on a truck, HMMWVs can embark on an LCAC, and fixed-wing aircraft can land on and take off from aircraft carriers.



To use embarkation in HLA federations, you must use RPR FOM 2.0, draft 17 or later. For information about specifying the correct FOM, please see "Starting VR-Forces," on page 5-3 and "Specifying the RPR FOM Version," on page B-16.

1.1.5. Mission Planning

You can create plans for entities that contain tasks, set data requests, conditional statements and global commands. You can save plans and edit them.

Conditional statements specify that tasks will be executed only if a specific condition is true or false. Some of the conditions tested are:

- Is an entity in a specific area?
- Is an entity destroyed?
- Is an entity to the left or right of a phase line?

In addition to conditional statements that are checked only once (`If` statements), VR-Forces supports `When` statements and `While` statements.

Global Plans

Global plans exist independently of any entity. They allow you to assign tasks and set data requests to entities and execute other commands. In essence, they allow you to script the sorts of ad hoc actions that a VR-Forces user could take through the GUI. This provides a great amount of flexibility to automatically modify the order of battle and tactical graphics in response to events.

For more information, please see Chapter 11, *Writing Plans*, in *VR-Forces Scenario Management Guide*.

1.1.6. Entity Tasks

You can assign tasks to simulation objects as part of a plan or independently. The tasks supported in entity-level scenarios include:

- Move to an object.
- Patrol along a route.
- Patrol between two objects.
- Follow an entity.
- Fire for effect.
- Take off and land.
- Fly to a heading.

The tasks supported in aggregate-level scenarios include:

- Move to an object.
- Patrol along a route.
- Patrol between two objects.
- Construct a minefield (and other combat engineering objects)
- Breach an obstacle.
- Bomb a location.

Independently assigned tasks override the tasks in a plan, allowing you to respond to the course of a simulation as events transpire, without stopping to edit an entity's plan. For more information, please see Chapter 7, *Assigning Tasks*, in *VR-Forces Scenario Management Guide*.

1.1.7. Scripted Tasks

You can create new tasks by writing scripts using the Lua scripting language. Input dialog boxes are created automatically and tasks are added to the Task menu. This feature requires some programming skills, but you do not need a developers license or a C++ development environment. For more information please see Chapter 16, *Creating Scripted Tasks*, in *VR-Forces Scenario Management Guide*.

Reactive tasks are a special kind of scripted task. They monitor the simulation and get executed only if some special condition is met. You can manage assignment of reactive tasks in the GUI. For more information, please see Section 7.3, "Reactive Tasks," in *VR-Forces Scenario Management Guide*.

Background processes are another special type of scripted task. They run continuously and manage ongoing entity activities. There is no user interface for managing them. For more information, please see Section 17.11, "Background Processes," in *VR-Forces Scenario Management Guide*.

You can organize scripted tasks into groups called Behavior Sets. By assigning Behavior Sets to forces you can control whether or not tasks are available to entities in those forces. For more information, please see Section 7.4, "Using Behavior Sets to Manage Scripted Tasks," in *VR-Forces Scenario Management Guide*.

1.1.8. Tactical Graphics

Using VR-Forces, you can interactively create and name tactical graphics, such as:

- Points
- Phase lines
- Lines
- Areas
- Obstacles.

Entities are aware of some types of tactical graphics and can be assigned tasks such as following a route, or moving between points. VR-Forces can also determine if specified entities are in an area and this information can be used as part of mission planning.

Aggregate-level scenarios support combat engineering objects, which can be areal, point, or linear objects that could impede, damage, or defend an entity.

Other types of tactical graphics are for informational and tactical management purposes and do not affect the simulation. For more information, please see Chapter 6, *Overlays and Tactical Graphics*, in *VR-Forces Scenario Management Guide*.

1.1.9. Terrain Agility and Composability

VR-Forces allows you to build your terrain at runtime using a variety of database and vector formats. It can load OpenFlight DTED, CTDB, MetaFlight, and GDB (MÄK terrain format) databases. It can load feature data from shapefiles, S-57 files, and GDB files. You can also superimpose geospatial images such as GeoTiffs or other raster image files on the terrain.

You can load local terrain data and you can stream data from networked servers using the osgEarth earth file format.



VR-Forces only supports geocentric terrains in osgEarth earth files.

You can save these composed terrains in the MTF file format. For details, please see Chapter 13, *Composing Terrains*, in *VR-Forces Configuration Guide*. For a list of supported file formats, please see Section 3.12.2, "Supported File Formats," in *VR-Forces Configuration Guide*.

VR-Forces supports the following coordinate systems:

- UTM
- Geocentric
- MGRS
- Latitude, longitude
- Database.

You can overlay raster images, such as CADRG, BMP, and geotiff files on terrain databases.

1.1.10. Flexible, Intuitive Graphical User Interface

VR-Forces's graphical user interface (GUI) allows you to manipulate VR-Forces from menus, dockable control panels, and the keyboard.

The VR-Forces display includes the following features:

- Manipulate the simulation map by zooming and panning.
- Select and track entities.
- Turn display features on and off.
- Display a hierarchical, echelon view of entities and an embarkation view.
- Display entity labels that include several types of information.
- Display line-of-sight (intervisibility) for entities.

You can hide and display the control panels, keep them docked to the screen, or undock them to make more space available for the display window. The GUI controls can be hidden for full-screen demonstrations.

The GUI follows standard conventions for modern windowing systems. For additional details, please see Appendix A, *Common GUI Controls and Behaviors*. For descriptions of individual windows and dialog boxes, please see online help.

1.1.11. Overlays

VR-Forces supports overlays. Overlays mimic clear plastic overlays that a planner might use to draw graphics on top of a terrain map. All objects are automatically assigned to overlays. You can add overlays and move objects between overlays. You can hide the display of objects on an overlay.

1.1.12. Behaviors

Entities understand how to interact with their environment. They can:

- Respond appropriately to other entities and objects, for example, avoiding entities, waiting for orders, and so on.
- Respond to detected targets, for example, deciding which of multiple targets to acquire, which weapon to fire, and what ammunition to use.
- Follow a route.

1.1.13. Multiple Attach Modes

VR-Forces offers attach modes that let you:

- Manually control the observer's position and orientation. (The observer represents the user's view, or eyepoint, in the simulated world.)
- Follow an entity, maintaining a consistent positional offset, with a matching heading.
- Mimic a vehicle's position and orientation. You can place the observer inside the vehicle for a driver's-eye view, or fly outside of it, moving in tandem with the vehicle.
- Tether the observer to an entity, maintaining a consistent positional offset without changing the observer's heading.
- Automatically track a vehicle's movement from a fixed viewpoint, as if watching from a control tower.

For more information, please see Chapter 11, *Moving the Observer*.

1.1.14. Special Effects and Entity Information Visualization

VR-Forces includes features that help you understand the interaction between entities in a simulation and others that add a realistic tone to interactions. These features include:

- Fire and detonation lines.
- Track histories.
- Range rings.
- Radio communications lines.
- Trailing effects such as footprints, tire tracks, and wakes.

1.1.15. Dynamic Ocean

VR-Forces provides dynamic ocean visualization in the Stealth and IG observer modes. The ocean shows waves, swells, and spray effects. Surface entities have realistic wakes and buoyancy behavior. Changes to wind speed and direction affect wave action.

1.1.16. Lighting Effects

VR-Forces supports a variety of lighting effects:

- Dynamic lighting
- Ocean planar reflection
- Lens flare
- Crepuscular (sun) rays.

Shader-based Effects Maps

VR-Forces supports several types of shader-based effects texture maps. Shaders are computer programs that simulate the effect of light on the terrain and objects in the simulation. VR-Forces makes extensive use of them for visualization effects.

VR-Forces supports the following types of shader-based effects maps:

- Normal, or bump, maps. Normal maps give terrains the appearance of relief, such as a rocky landscape.
- Specular maps. Specular maps affect the highlight color of objects.
- Ambient occlusion maps. Ambient occlusion maps model areas that do not receive direct light, such as cracks and crevices and shaded areas of terrain and models. These areas are lit only by ambient light.
- Reflection maps. Reflection maps affect the reflectivity of surfaces, such as windows. Reflection maps reflect objects in the sky, not the terrain.

Lighting effects are described in Chapter 17, *Lighting Effects*.

1.1.17. Accurate Vehicle Positioning

VR-Forces can smooth the trajectories of moving vehicles to compensate for discontinuous positional data. The ground clamping feature can ensure that all ground entities are placed correctly on the terrain surface. For details, please see "Trajectory Smoothing," on page 8-5 and "Using Ground Clamping," on page 8-3.

1.1.18. Batch Mode Operation

Batch mode allows you to run a scenario multiple times or run several different scenarios, without direct action through the graphical user interface. This is useful for running unattended testing or to test the results of a scenario under differing conditions. You can record the results of batch mode simulations using the MÄK Logger, for later review. For more information, please see Section 1.6, "Running VR-Forces in Batch Mode," in *VR-Forces Scenario Management Guide*.

1.1.19. Remote Control

You can control the VR-Forces observer remotely through view control messages, which can be generated by MÄK applications such as MÄK Data Logger, and by programs that use the VR-Vantage Control Toolkit. For details, please see "Controlling the Observer from Other Applications," on page 11-27.

You can also send view control messages from entity plans and global plans. For details, please see Section 11.6.6, "Sending View Control Messages," in *VR-Forces Scenario Management Guide*.

1.2. The VR-Forces Toolkit

The VR-Forces toolkit lets you develop CGF applications. You can enhance the capabilities of the VR-Forces simulation engine and GUI, or integrate the simulation engine into your own user interface. The toolkit has several APIs. The Simulation API gives you control over:

- Behaviors
- Components
- Entity types
- Parameters
- Messages
- Resources
- Tactical graphics
- Plans
- Tasks.

The Remote Control API allows you to control a VR-Forces simulation engine from a remote application, such as a graphical user interface (GUI). The Terrain API lets you manipulate terrain databases.

The VR-Forces front-end is built with the VR-Vantage Toolkit. The toolkit is included with VR-Forces, allowing you to modify or extend the front-end. For details about the VR-Forces Toolkit, please see the HTML API documentation.

1.2.1. Plug-in Architecture

You can modify the VR-Forces front-end and back-end by rebuilding the applications or by writing plug-ins. In fact, much of the basic functionality of the front-end is implemented using plug-ins rather than as part of the main executable. Plug-ins are dynamically linked libraries that VR-Forces can load at startup. The rationale for choosing which method to use is described in API documentation. You can manage the use of plug-ins in the front-end. For details, please see "Managing Plug-ins," on page 5-32.

1.3. The B-HAVE Module for VR-Forces

The B-HAVE (Brains for Human Activity in Virtual Environments) Module for VR-Forces is a VR-Forces plug-in that extends the movement models for lifeforms and ground platforms, as follows:

- Full 3D path finding.
- Simplicity of use of basic agents (GoTo).
- Complexity and realism of behaviors (Hide, Flee, Wander).
- Complexity and realism of scenarios.

The B-HAVE Module is powered by Kynapse, the world's leading AI technology solution, from Autodesk, Inc. It is based on Kynapse's fundamental concepts of Topological Modeling for Path Finding, and Spatial Reasoning for Advanced Behaviors, all integrated within VR-Forces as advanced planning capabilities.

The B-HAVE Module offers VR-Forces users the ability to:

- Make entities move autonomously without specifying routes.
- Make entities navigate through complex environments, indoors and outdoors, while avoiding collisions with obstacles and other entities.
- Quickly populate a 3D environment with dozens of entities wandering in buildings, on roads, and on pavement.
- Develop complex scenarios in which entities hide from each other and flee from threats.

The B-HAVE Module is a separately licensed product. It is not included with VR-Forces. Please see your MÄK salesperson for details.

1.4. DI-Guy

VR-Forces uses DI-Guy software and content for human character animation.

VR-Forces comes with DI-Guy functionality built-in and with a large set of DI-Guy characters, appearances, and animations. A VR-Forces customer does not need a DI-Guy license of any kind to use DI-Guy animated characters in a VR-Forces-based application (including custom applications built using the VR-Forces Toolkit).

However, you need a DI-Guy Developer's license if you want to:

- Develop new character types, or alter the geometry, appearances, textures, or motions of DI-Guy characters.
- Directly access the DI-Guy API to modify or extend the way that VR-Forces uses the DI-Guy API or access DI-Guy functionality that MÄK has not already integrated into VR-Forces.

DI-Guy libraries, models, textures, motion files, and other graphics resources are distributed in the VR-Forces package solely so that VR-Forces can use them. Use of DI-Guy software or content outside of VR-Forces-based applications is not permitted by the VR-Forces license.

For more information, please see Section 8.12, "DI-Guy Animation," and Section 10.18, "DI-Guy Appearance," in *VR-Forces Scenario Management Guide*, and Section 2.3, "Configuring DI-Guy Integration," and Section 6.2.5, "Editing a Lifeform's DI-Guy Character, Appearance, and Animation," in *VR-Forces Configuration Guide*.

1.5. Support for External Communications Servers

You can configure VR-Forces to use an external communications effects server to control the transmission of radio messages between VR-Forces entities. The communications effects server determines which entities are able to communicate with each other, and how long it takes a message to be transmitted from the originator of the message to the receiver of the message.

Any communications effects server that meets the defined interface can be used. For information about the interface, please contact support@mak.com. For more information, please see Chapter 11, *Using an External Comms Effects Server*, in *VR-Forces Configuration Guide*.

1.6. Helpful Utilities

VR-Forces includes the following utilities for configuring and troubleshooting VR-Forces:

- The Entity Editor is a graphical user interface that lets you add and remove component systems from entities and specify common entity parameters. You can create new entities in the editor. You can also create new simulation model sets. The Entity Editor is also used to manage tactical graphics. For details, please see Chapter 5, *Introduction to the Entity Editor and SMSs*, in *VR-Forces Configuration Guide*.
- The Object Parameter Database (OPD) Editor is a graphical user interface for editing the object parameter database. Using the OPD Editor can speed up editing of the object parameter database and prevent syntactical errors that can occur when you use a text editor to edit parameters. For details, please see Chapter 9, *Using the OPD Editor*, in *VR-Forces Configuration Guide*.
- The Scenario Merge tool lets you merge scenarios. This allows development teams to compose portions of a large scenario individually and easily merge them into one combined scenario. For details, please see Chapter B, *Merging Scenarios*, in *VR-Forces Scenario Management Guide*.
- The vrfMsgDump utility displays the contents of VR-Forces messages discovered on the network.
- The Boundary Tool lets you calculate the boundaries of a high resolution terrain inset for inclusion in an *.earth* file. For details, please see Chapter 5, *Introduction to the Entity Editor and SMSs*, in *VR-Forces Configuration Guide*.
- The medfTool lets you encrypt 3D model files. For details, please see Section 16.3, "Compressing Model Files," in *VR-Forces Configuration Guide*.
- The mftTool lets you process MetaFlight terrains so that they will work in VR-Forces. For details, please see Section G.3.1, "The mftTool," in *VR-Forces Configuration Guide*.
- osgearth_cache. This utility lets you cache terrain data from earth files offline. For details, please see Section 4.10.2, "Caching osgEarth Terrain Data Offline," in *VR-Forces Configuration Guide*.

1.7. Third-Party Software and Content

The VR-Forces front-end includes software and content licensed from third parties, including:

- SilverLining™: real-time sky and 3D cloud rendering from Sundog Software.
- GL Studio®: interactive cockpit instrumentation and HMI from DiSTI®.
- SpeedTree®: animated, 3D foliage from Interactive Data Visualization (IDV).
- 3D models, terrain, and graphical content from Simthetiq, RealDB, TurboSquid, Terrex (now Presagis), and TerraSim.
- OpenSceneGraph: an open source 3D graphics toolkit hosted at http://www.openscenegraph.org.
- osgEarth: an open source streaming terrain plug-in by Pelican Mapping, at http://osgEarth.org.

! The run-time and developers' rights for VR-Forces customers vary from vendor to vendor. Please read this section carefully to understand which rights are included with VR-Forces licenses, and which rights require additional third-party licenses.

1.7.1. SilverLining

VR-Forces uses SilverLining software and content to compute lighting and to render the sky, clouds, sun, and moon. SilverLining is developed by Sundog Software (http://www.sundog-soft.com).

Most VR-Forces customers do not need to buy a SilverLining license of any kind. In general, a VR-Forces customer has the right to use the SilverLining technology that is built into VR-Forces in any VR-Forces-based application (including custom applications built using the Toolkit). However, there are two exceptions:

- If you want to extend or change the way that MÄK has integrated SilverLining into VR-Forces by writing directly to the SilverLining API, then you need a SilverLining Developer's License.
- If you are building a commercial video game or other "mass-market" (consumer-level) commercial product on top of the VR-Forces Toolkit, and the resulting product includes the SilverLining functionality, then you need a SilverLining Developer's License.

SilverLining libraries, textures and other graphics resources are distributed in the VR-Forces package solely so that VR-Forces can use them. Use of SilverLining software or content outside of VR-Forces-based applications is not permitted by the VR-Forces license.

1.7.2. GL Studio

VR-Forces uses GL Studio software and content to render interactive cockpit instrumentation displays. GL Studio is developed by DiSTI (http://www.disti.com).

VR-Forces is delivered with several generic cockpit displays. You can use the built-in cockpit displays in any VR-Forces-based application (including custom applications built using the Toolkit), without a GL Studio license of any kind.

If you want to build custom cockpits using the GL Studio editor or GL Studio API, you need a GL Studio Developer's license. If you want to execute custom cockpits in VR-Forces-based applications (regardless of whether the custom cockpits are built by you or a third party), you need GL Studio Run-Time licenses.

GL Studio libraries and graphics resources are distributed in the VR-Forces package solely so that VR-Forces can use them. Use of GL Studio software or content outside of VR-Forces-based applications is not permitted by the VR-Forces license.

1.7.3. SpeedTree

VR-Forces uses SpeedTree software and content for animated real-time 3D foliage and vegetation. SpeedTree is developed by Interactive Data Visualization (IDV) (http://www.speedtree.com).

A VR-Forces customer does not need a SpeedTree license of any kind to use the SpeedTree functionality that is built into the VR-Forces applications that MÄK delivers. You may build and run plug-ins to our out-of-the-box applications without a SpeedTree license, as long as your plug-ins do not need to directly access the SpeedTree API.

However, you need a SpeedTree Developer's license if you want to:

- Write a plug-in that uses the SpeedTree API directly to extend or modify the way that SpeedTree is integrated into VR-Forces.
- Use the VR-Forces Toolkit to develop new applications, and you want your applications to include SpeedTree functionality.

SpeedTree libraries, models, textures, and other graphics resources are distributed in the VR-Forces package solely so that VR-Forces can use them. Use of SpeedTree software or content outside of VR-Forces-based applications is not permitted by the VR-Forces license.

1.7.4. 3D Models, Terrain, and Graphical Content

VR-Forces includes a rich set of 3D models for vehicles, weapons, cultural features and urban clutter (signs, barriers, lampposts, and so on). It also includes several sample terrain databases to help you get started with VR-Forces, and to help demonstrate VR-Forces. Much of this content is derived from 3D data that we have licensed from third parties:

- Many of the high-quality vehicle models come from Simthetiq (http://www.simthetiq.com).
- Some models come from Real DB (http://www.realdb.qc.ca/company).
- Some of the middle-eastern building models and urban clutter objects that are used in our sample terrain databases were licensed from Turbosquid (http://www.turbosquid.com).
- The TerraSim_SampleUrban terrain database comes from TerraSim (http://www.terrasim.com).
- The HarrisAfghanVillage terrain database was created using source data from Harris Inc.
- The Berkeley Database was developed by Terrex (now Presagis) (http://www.presagis.com).

All of this content is distributed in the VR-Forces package solely so that VR-Forces can use it. Use of any of the VR-Forces models, textures, terrains, or other content outside of VR-Forces-based applications is not permitted by the VR-Forces license.

1.7.5. OpenSceneGraph

VR-Forces uses OpenSceneGraph for its underlying scene graph representation, rendering, file loading, and many other capabilities. OpenSceneGraph is an open source, cross-platform graphics toolkit for the development of high-performance graphics applications. The OpenSceneGraph source repository is maintained by Robert Osfield at http://www.openscenegraph.org/projects/osg. It is distributed under the OpenSceneGraph Public License (OSGPL), which is based on the GNU Lesser General Public License (LGPL).

MÄK will provide modified source code for OSG upon request, as per the OSGPL license. Please contact support@mak.com to obtain the download links for our modified source.

1.7.6. osgEarth

VR-Forces uses osgEarth to import streaming terrain elevation and imagery data. The data can be streamed from external servers and sources or from a directory on the computer running VR-Forces. osgEarth is an open source plug-in to OpenSceneGraph, maintained by Pelican Mapping at http://osgEarth.org. osgEarth is distributed under the GNU Lesser General Public License (LGPL).

MÄK will provide modified source code for osgEarth upon request, as per the OSGPL license. Please contact support@mak.com to obtain the download links for our modified source.

1.8. Distributed Simulation Standards Supported

VR-Forces supports the HLA 1.3 specification, the current draft of the IEEE 1516 C++ API maintained by the SISO Dynamic Link Compatible RTI API Product Development Group, and HLA Evolved (IEEE 1516-2010). It also supports the Distributed Interactive Simulation (DIS) protocol, versions 4, 5, 6, and 7.

VR-Forces has built-in support for the HLA RPR-FOM and can support other FOMs through the FOM Mapping feature. For information about FOM mapping, please see "Specifying a FOM Mapper," on page B-16. Please see *VR-Forces Release Notes* for a list of the versions of DIS supported.

VR-Forces supports time management for HLA exercises.



2. Installing VR-Forces

This chapter explains how to install VR-Forces on your computer. It also explains how to install other software required for use with VR-Forces.

Installing VR-Forces 2-2
- Installing VR-Forces on Windows 2-2
- Installing VR-Forces on a Linux System 2-3
- Uninstalling VR-Forces on Windows 2-3

The VR-Forces Directory Structure 2-4

Installing and Setting Up the MÄK License Manager 2-5
- Specifying the License Server 2-6

Installing an RTI 2-9
- Installing the MÄK RTI 2-10

Localizing the Graphical User Interface 2-11
- Translating Other Interface Files 2-14
- Translating VR-Forces Scripts and Console Messages 2-15
- Applying the Language Files 2-16
- Merging Translation Files 2-16

2.1. Installing VR-Forces

A full VR-Forces installation includes three components: the application, the License Manager software, and for HLA operation, an RTI (or Runtime Infrastructure.)

Before you install VR-Forces, please read the Release Notes to see if there are any special instructions for installation.

VR-Forces has two installers, an application installer and a data installer. You must run both installers. You must run the application installer first.

To install the VR-Forces toolkit, you must enter an installation code. Be sure you have received your code from your MÄK salesperson before you start the installation process. If you do not have an installation code, you can install the runtime files. When you receive your installation code you can install the toolkit.

2.1.1. Installing VR-Forces on Windows

Note the following:

- You must have administrator privileges to install MÄK products on Windows Vista.
- When you install large applications on Windows Vista or later, 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-Forces 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-Forces and whether it is a 32 bit or 64 bit version.

➤ To install VR-Forces, run the installers. Follow the instructions in the installation wizards. (You must run the application installer and the data installer.)

2.1.2. Installing VR-Forces on a Linux System

Linux versions of VR-Forces are provided as compressed tar files on DVD, or as downloaded files.



If you install VR-Forces in one directory and user data in another directory, VR-Forces will not start.

To install VR-Forces on Linux:

1. Create the directory in which you want to install VR-Forces.
2. Copy the tar file to the install directory.
3. Uncompress and untar the application file:

   `tar -vxzf application.tar.gz`

   where *`application`* is the product release identification.
4. Uncompress and untar the data package into the same directory as the application.

2.1.3. Uninstalling VR-Forces on Windows

Because installation of VR-Forces is a two step process, uninstallation is also a two-step process.

To uninstall VR-Forces on Windows:

1. In the VR-Forces root directory, run *unins001.exe*.
2. In the VR-Forces root directory, run *unins000.exe*.

2.2. The VR-Forces Directory Structure

The VR-Forces installation process creates a directory structure like the one shown in Figure 2-1. On PCs, the default installation directory is *C:\MAK\vrforces4.3.1*.)

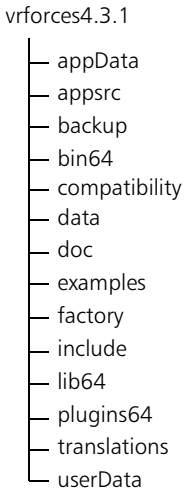


Figure 2-1. VR-Forces directory structure

Table 2-1 describes the contents of the directories installed by VR-Forces.

Table 2-1: Contents of VR-Forces directories (Sheet 1 of 2)

Directory	Directory contents
./appData	Schema, model definitions, settings, configuration files, and other data.
./appsrc	Source files and project files for API developers.
./backup	Backup configuration files.
./bin64 or *./bin*	Product executables and DLLs.
./compatibility	Header files for compatibility with releases before VR-Forces 4.3.
./data	The *./data* directory has several subdirectories that contain: ◆ Bitmaps and icons. ◆ Simulation model sets. ◆ Terrain database source files. ◆ Visual model data.
./doc	VR-Forces documentation.
./examples	Sample code for toolkit users.
./factory	Backups of the application data and settings provided by MÄK.
./include	Header files for use by API developers.
./lib64 or */lib*	Library files.

Table 2-1: Contents of VR-Forces directories (Sheet 2 of 2)

Directory	Directory contents
./plugins64 or *./plugins*	Plug-in DLLs.
./translations	Files and an executable for localizing the GUI.
./userData	Scenarios, terrain files and other files that get created by users as they use VR-Forces.

2.3. Installing and Setting Up the MÄK License Manager

Before you can use a MÄK product, you must obtain a valid license file, install the MÄK 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 MÄK 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 MÄK 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 MÄK installation media. It is separate from the product installers. You can download the installers from our web site at http://www.mak.com/license-support.html or you can download directly from:

- Windows:
  ftp://ftp.mak.com/out/MAKLicenseManager-win-setup.exe (for 32-bit systems)
  ftp://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe
- Linux:
  ftp://ftp.mak.com/out/MAKLicenseManager-linux-setup.tar.gz (for 32-bit systems)
  ftp://ftp.mak.com/out/MAKLicenseManager-linux64-setup.tar.gz

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.

2.3.1. Specifying the License Server

The first time you run a MÄK application on a particular computer, the License Setup dialog box opens (Figure 2-2). It prompts you to enter the hostname of the license server and optionally, a port number.

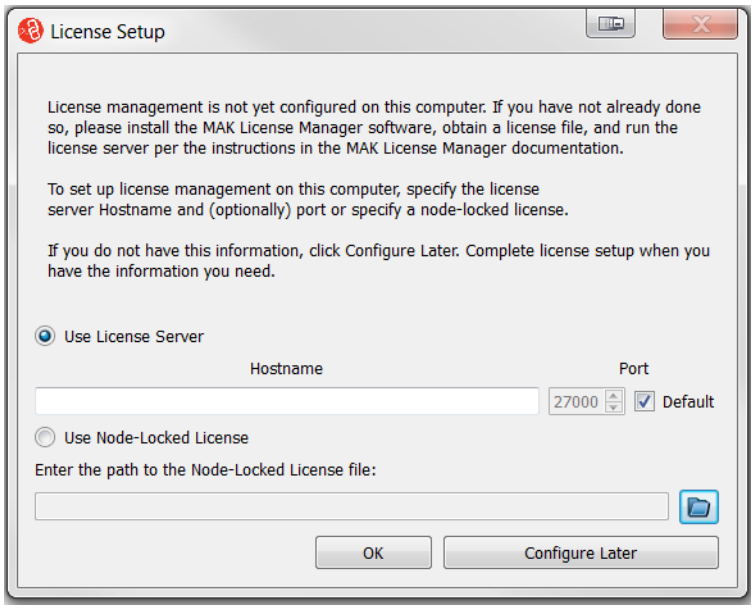


Figure 2-2. License Setup dialog box

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 MÄK applications until you set up license management.

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

Under limited circumstances, MÄK 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 MÄK 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 MÄK application.
- You can clear the saved license configuration by deleting the cache file. On Windows XP, it is *C:\Documents and Settings\user_name\Application Data\MAK\license<n>.xml*. On Windows 7, it is *C:\Users\user_name\AppData\Roaming\MAK\licenses<n>.xml*. (The *AppData* directory is hidden by default.) On Linux, it is *.mak/license<n>.xml*. (There may be more than one cache file, for example, *licenses1.xml* and *licenses2.xml*.)

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`.

The following sections explain how to set environment variables on the different platforms that MÄK products run on.

Windows

To add the MAKLMGRD_LICENSE_FILE in Windows:

1. Open the Control Panel.
2. On Windows 7, click System and Security.
3. Click System.
4. In the sidebar menu, click the Advanced System Settings. The System Properties dialog box opens.
5. Click Environment Variables. The Environment Variables dialog box opens.
6. Click New. The New System Variable dialog box opens.
7. In the Variable Name field, enter MAKLMGRD_LICENSE_FILE.
8. In the Variable Value field, enter `@server_name`, where `server_name` is the name of the license server.
9. Click OK to back out of each dialog box and set the variable.



Different versions of Windows have slightly different wording of the various Control Panel options.

Linux

On Linux, you set environment variables in your *.cshrc* (or equivalent startup file). Set the variable similarly to the following example:

```
setenv MAKLMGRD_LICENSE_FILE @oak
```

If you are using the sh or bash shells, you set environment variables in your *.profile* file (or *.bashrc*). Set the variable similarly to the following example:

```
MAKLMGRD_LICENSE_FILE=@oak
export MAKLMGRD_LICENSE_FILE
```

Do not put spaces around the equal (=) sign.

You are ready to run the license server and use your new licenses or MÄK products.

2.4. 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 MÄK products, please see the release notes for your MÄK application.

2.4.1. Installing the MÄK RTI

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

Configuring Your System to Use the MÄK RTI

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

- Linux — LD_LIBRARY_PATH.
- Windows — PATH.

The MÄK 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 MÄK RTI

To run a MÄK application with the MÄK 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 MÄK RTI, however you can run the rtiexec if you want to. (It is required to use certain features of the MÄK RTI.) For more information, please see your RTI documentation.



If you are running VR-Forces, it is recommended that you run the rtiexec.

2.5. Localizing the Graphical User Interface

You can localize your MÄK product's graphical user interface so that it uses languages other than English. Your product includes a file called *./translations/product_untranslated.ts*, for example *logger_untranslated.ts* or *vrforces_untranslated.ts*. The file contains entries for all of the application-specific menu and dialog box text. The Qt Linguist tool, which is included with VR-Forces, lets you map the English text provided by MÄK to your chosen language.



The *untranslated.ts* file does not include all of the text that may be displayed in the GUI or in console and error messages.

Some strings, such as those on the Create menu, are in configuration files. Others are in settings files. You can translate these strings by identifying the appropriate configuration file and translating the strings.

VR-Forces has a utility that can scan system scripts and SMS files and create translation files for user-visible text. For details, please see "Translating VR-Forces Scripts and Console Messages," on page 2-15.

To edit a resource file:

1. Start the Qt Linguist utility. It is *./translations/linguist.exe*
2. In the Qt Linguist window, choose **File** → **Open**. Select the file *./translations/product_untranslated.ts*. The Settings for filename dialog box opens (Figure 2-3).



Figure 2-3. Linguist settings dialog box

3. Select values for the source language, target language, and target region or country.
4. Click OK. The translation file is opened and a list of classes is displayed on the left side of the window (Figure 2-4).

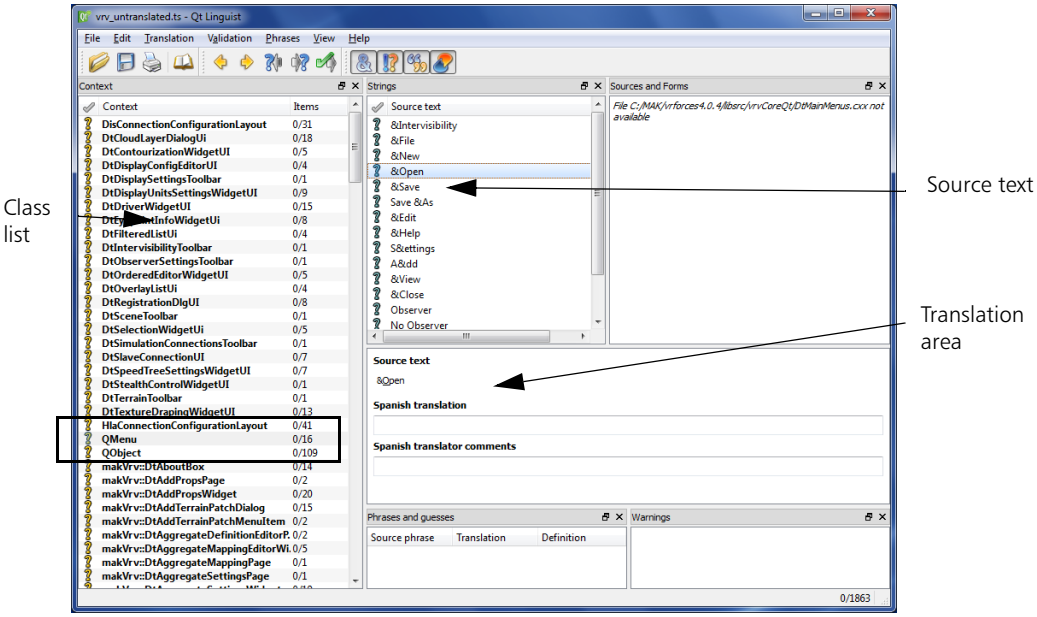


Figure 2-4. Linguist window

5. To translate text, you select each class in turn and translate the text associated with that class. You do not have to understand anything about what the classes mean, you just have to translate the text. In Figure 2-4, *QMenu* is selected. The Strings window lists text that can be translated. Text that has not been translated has a question mark in the checkmark column.

6. In the Source Text list, select the word you want to translate. The text is displayed under Source Text in the translation area (the window below the source text list.) Figure 2-5 shows the text Open selected.

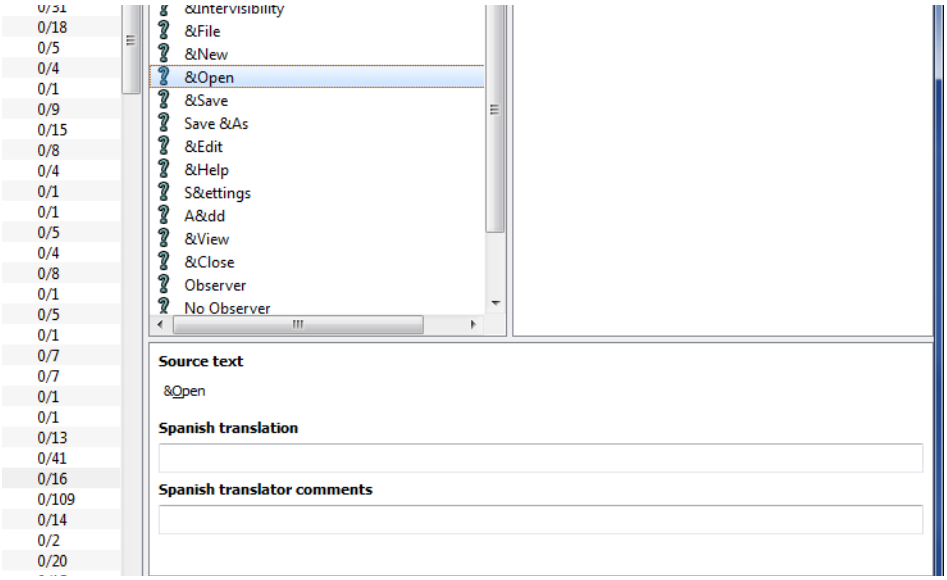


Figure 2-5. Text selected for translation

7. Place the cursor in the text box below *Language* Translation, where *Language* is the target language that you selected when you opened the translation file.
8. Type the translation text.
9. When you are done and you are certain of the translation, select the Done and Next button on the toolbar. A green check mark replaces the question mark (Figure 2-6). If you move to a different text string without clicking Done and Next, a yellow question mark is displayed for the string you translated. When you translate all elements for a class, a green check mark replaces the question mark next to the class name.

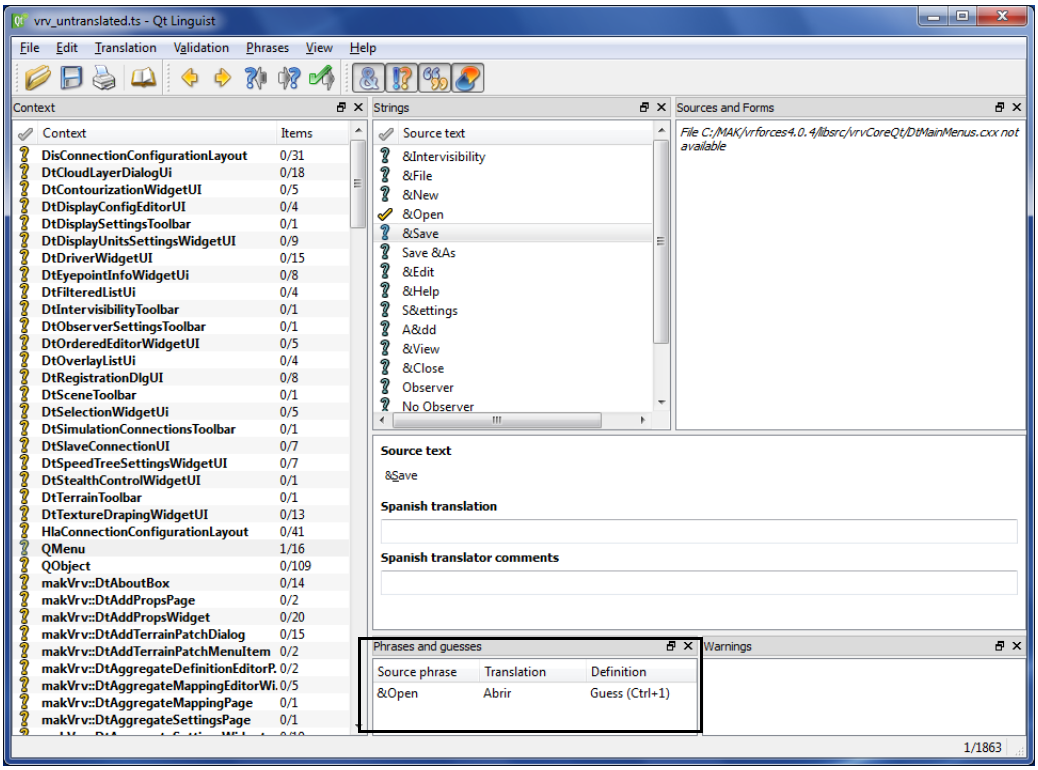


Figure 2-6. Completed translation for Open

10. Repeat this process for each element of source text for each class.
11. Choose **File** → **Release**. The file is saved in the format *filename.qm*.
12. Rename the file to a name that indicates the language to which you have translated the file, using the two-character abbreviation typically used with your language. For example, a Spanish translation would be renamed *product_es.qm*.
13. Save your translated *.ts* file.

2.5.1. Translating Other Interface Files

The *product_untranslated.ts* file contains all the text used in the VR-Forces interface. However, there may be other text and messages that are not specific to VR-Forces that are generated by the software that implements the VR-Forces GUI. To translate this text, you need to translate the file *qt_untranslated.ts*. Follow the same procedure as you did for *vrforces_untranslated.ts*.

2.5.2. Translating VR-Forces Scripts and Console Messages

The *vrforces_untranslated.ts* file includes text for dialog boxes and interface elements created using the Qt Toolkit. However, there are many messages generated by code and the dialog boxes for scripted tasks are generated programmatically, rather than through Qt ui files. Therefore they are not included in *vrforces_untranslated.ts*. VR-Forces has a utility that can scan all system scripts (those in a simulation model set) as well as some of the common SMS files, and create a TS file for each script and SMS. You can then translate the text in the individual TS files. This script also creates a TS file for DI-Guy files.

For this process to work, you must code messages in your scripts in a certain way. For details, please see Section 17.6, "Coding Messages for Translation," in *VR-Forces Scenario Management Guide*. VR-Forces also has a way to code console messages so that they can be extracted and translated. For details, please see 17.8.1, Coding Object Console Messages for Translation, in *VR-Forces Developers Guide*.

Generating TS Files for Scripts and Simulation Model Sets

If you code your scripts to support translation, you create TS files by running the translationFileCreate utility. By default this application scans *./data/simulationModelSets* and its subdirectories for all system scripts and some generic files, and *./appData/settings* for DI-Guy files.

The utility creates an *untranslated.ts* file for each lua script. This file is saved in the directory that contains the script. It also creates a TS file for each simulation model set, which is saved in the SMS directory and a *diguy_untranslated.ts* file, which is saved in *./translations*.

When you translate a script file, the translated file must remain with the back-end(s) that are hosting the scripts. The rest of the translated files must exist on the user machine. The .qm file for DI-Guy and VR-Forces must live in the translations directory and the .qm file for the simulation model sets with the .sms files.

The syntax for translationFileCreate is as follows:

```
translationFileCreate.exe  [--directory filename] ...
   [--smsfile string] ...  [--nodiguy] [--nosms]
   [--noscripts] [--] [-v] [-h]
```

Table 2-2: translationFileCreate arguments

Argument	Description
`--directory` *`directory_name`*	Directory to scan for files of any of the supported types (lua, sysdef, xml, mtl) as per creation rules (scripts, systems, supporting). Accepted multiple times.
`(-h | --help)`	Displays usage information and exits.
`--nodiguy`	Do not parse the DI-Guy configuration files.
`--noscripts`	Do not parse Lua files for translation.

Table 2-2: translationFileCreate arguments

Argument	Description
`--nosms`	Do not parse the simulation model set directories for their supporting translatable items.
`--smsfile` *`SMS_filename`*	Specific SMS file to use when translating simulation model set directories. Accepted multiple times. If not specified, processes all SMS files.
`(-v | --version)`	Displays version information and exits.
`(-- | --ignore_rest)`	Ignores the rest of the labeled arguments following this flag.

Generating TS Files for Object Console Messages

If the messages in your source code are coded for extraction to TS files, you can generate an untranslated.ts file for them. To generate TS files, run:

```
lupdate source_directory -ts my_new_translations.ts
```

The resulting TS file is saved to the location pointed to by the path given to `-ts`.

2.5.3. Applying the Language Files

To put the language file into effect, use one of the following methods:

- Set your computer's environment to the proper language (LANG environment variable).
- Set the locale to the proper language.
- Start VR-Forces with the `-G` *`locale`* command line option, where *`locale`* is the two-letter country code for the local language.

2.5.4. Merging Translation Files

If you localize a version of VR-Forces and upgrade to a newer version, you can merge your localized *.ts* file with the *.ts* file from the new version. Then you only have to translate new or changed text in the new version. You merge translation files with the *translationMerge* utility, which is in the *./bin* directory. The syntax is as follows:

```
translationMerge [-o] [-m] previous.ts current.ts new.ts
```

where:

- *`previous.ts`* is the name of the previously localized translation file.
- *`current.ts`* is the name of the translation file provided with the current release of VR-Forces.
- *`new.ts`* is the name to give to the merged file.
- `-o` specifies that the merged file removes obsolete messages.
- `-m` merges modules that have equal text strings.



3. VR-Forces Simulation Concepts

This chapter describes at a conceptual level the components of a VR-Forces simulation and how they work together. *VR-Forces Scenario Management Guide* explains how to create and run simulations.

The VR-Forces Program 3-3
Front-end and Back-end Concepts 3-3
How Front-ends and Back-ends Work Together 3-4
How VR-Forces Back-ends are Identified 3-5
VR-Forces Sessions 3-5
Coordinating Multiple Front-ends 3-7
Working with Multiple Back-ends 3-8
Scenarios 3-9
Objects 3-10
The Object Parameter Database 3-10
Local Objects and Remote Objects 3-10
Entity-Level Modeling and Aggregate-Level Modeling 3-11
Entity and Aggregate Terminology 3-12
Entities 3-13
How Entities are Identified 3-13
How Entities are Organized 3-15
How Entities Communicate 3-16
VR-Forces Uses a Component Architecture 3-17
Entity Behaviors and Tasks 3-19
Aggregate Modeling 3-20
Aggregates in Entity-Level Scenarios 3-20
Creating Higher Echelon Aggregates in Aggregate-Level Scenarios 3-24

Tactical Graphics 3-25
Points 3-26
Phase Lines 3-27
Routes 3-27
Areas 3-27
Obstacles 3-28
Tasks 3-28
Plans 3-29
Set Data Requests 3-30
Terrain Databases 3-30
Composing Terrains (and Creating MTF Files) 3-31
Supported File Formats 3-31
Feature Layers and Props 3-33
Coordinate Systems 3-34
Loading Supported Databases 3-36
Using Large Terrain Databases 3-36
Intervisibility (Line-of-Sight) 3-38
Representing and Managing Time in VR-Forces 3-39
Simulation Time 3-39
Time of Day 3-39
Exercise Clock Modes 3-40
Using HLA Time Management 3-41
Interactions 3-41

3.1. The VR-Forces Program

Unlike most computer applications, which have one executable file, VR-Forces has two executable files: a front-end, or graphical user interface (GUI) and a back-end, or simulation engine. The front-end executable is vrfGui. The back-end executable is vrfSim-*protocol*, where *protocol* is DIS, HLA13, HLA1516, or HLA1516e.

By separating the simulation engine from the visualization program, VR-Forces allows you to run multiple front-ends, back-ends, or both. If you run multiple front-ends, several VR-Forces users can work together to create and manage scenarios. Using session IDs, you can specify which front-ends control which back-ends. The front-ends also display entities received over the network from other simulations participating in an exercise. (You have a limited ability to interact with these remote objects. For more information, please see "Local Objects and Remote Objects," on page 3-10, Section 11.9, "Writing Plans for Remote Entities," in *VR-Forces Scenario Management Guide*, and Section 10.12, "Configuring VR-Forces Components on Remote Entities," in *VR-Forces Configuration Guide*.)

If you run multiple simulation engines, the work of simulating entities and tactical graphics can be shared by multiple computers, which should improve performance. If you save a scenario, VR-Forces records which objects are simulated by each back-end, so that the configuration can be reproduced the next time you run the scenario.



In this manual, most procedures apply to interaction with the front-end. For the sake of simplicity, unless we are discussing a subject that applies specifically to either the front-end or back-end, we refer to VR-Forces in the sense of the combined interactions of both executables.

3.2. Front-end and Back-end Concepts

Although the separation of front-ends and back-ends, and the ability to run multiple instances of each adds to the power and flexibility of VR-Forces, you need to understand how the various executables interact to ensure that your scenarios are created and run smoothly. In this section, we describe:

- How front-ends and back-ends work together.
- How back-ends are uniquely identified.
- Use of sessions.
- How to coordinate multiple front-ends.
- How scenarios are loaded and saved when you run multiple back-ends.

3.2.1. How Front-ends and Back-ends Work Together

The VR-Forces front-end is an interface to the simulation engine. It communicates with the back-end by sending messages over the network. (This relationship exists even if you are not connected to a network.) It displays the state updates and interactions that the back-end generates and sends over the network. Figure 3-1 illustrates loading a scenario and running a simulation.

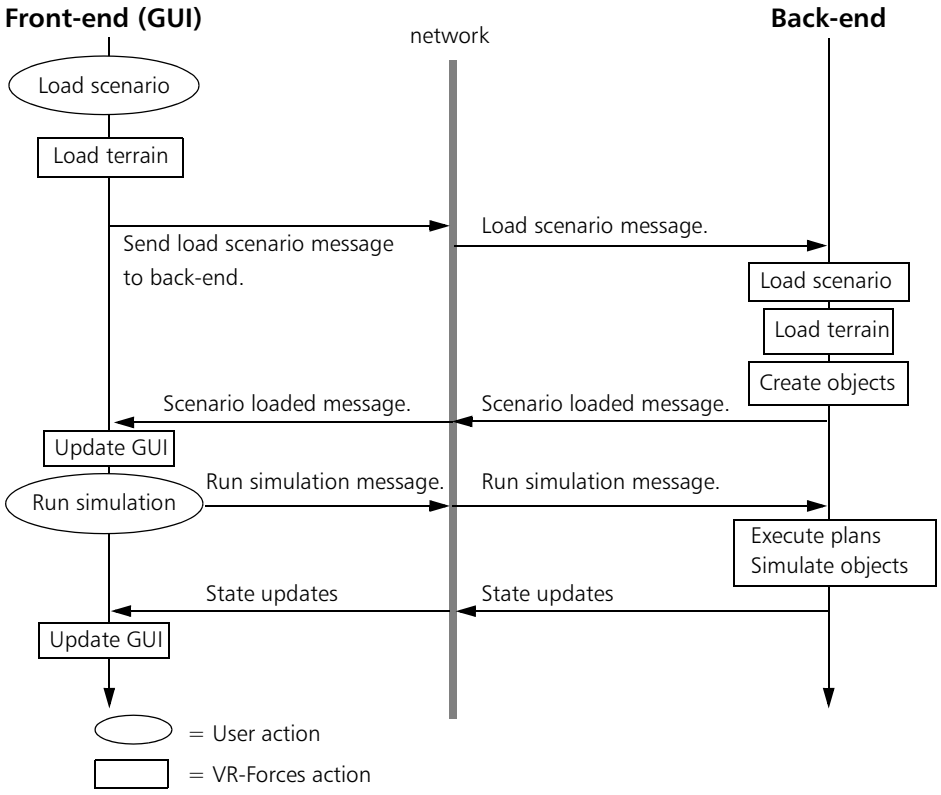


Figure 3-1. Front-end and back-end interaction

3.2.2. How VR-Forces Back-ends are Identified

If you run multiple back-ends, you can specify which back-end you want to use when you create entities and tactical graphics. This feature allows you to spread the simulation load across multiple PCs. However, to do this, each back-end must have a unique ID. VR-Forces back-ends are identified by their site ID and application number. Each back-end must have a unique site:application identifier. You assign the site:application identifier when you start an application. It is up to you to make sure that there are no duplicate identifiers. (For details, please see "Starting VR-Forces," on page 5-3.)



The site ID and application number concepts are borrowed from DIS. They apply to VR-Forces when it is running in HLA as well.

3.2.3. VR-Forces Sessions

VR-Forces uses sessions to help manage the relationships between front-ends and back-ends. Back-ends are always part of a session. Front-ends can join a session or operate independently of one. When a front-end is joined to a session, it can control the back-ends in that session. When a front-end is not joined to a session, it cannot control back-ends. However, it can open a terrain and view exercises running on its port. In this case it is just a viewer.

Use of sessions helps ensure consistency among front-ends that are controlling a common set of back-ends. The session mechanism ensures that:

- All front-ends and back-ends in a session run the same scenario and use the same terrain.
- When a front-end joins a session that is running a scenario, it automatically loads the correct terrain for the scenario.
- If any front-end in a session tries to create a scenario or load a scenario, and a scenario is already running, the user is warned that this action will destroy the currently running scenario.
- If any front-end in a session loads a scenario or creates a new scenario, all of the back-ends and other front-ends in that session take the same action.

Sessions provide the following additional benefits:

- They allow you to specify which front-ends control which back-ends. This feature could be useful, for example, if you want one set of VR-Forces applications to simulate friendly forces and another set to simulate opposing forces. If all applications were in the same session, participants could send orders to the entities in the opposing force. By using separate sessions, this is not possible.

  In Figure 3-2, the front-end with session ID 1 can control the back-ends with that same session ID. The front-end with session ID 2, controls the back-end with session ID 2.

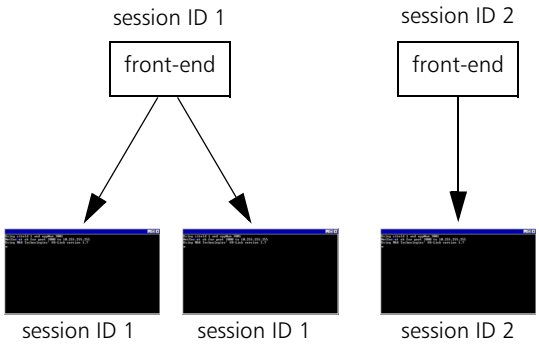


Figure 3-2. Using session ID to control back-ends

- Sessions allow you to load different terrain databases in different VR-Forces executables that are part of the same exercise. This feature may be helpful if you are participating in an exercise that covers a large geographical area, but only need to observe portions of that total terrain. Loading databases that cover a subset of the total exercise zone could improve the performance of your computer or the resolution of the terrain database.

  In Figure 3-3, the two sessions have loaded databases that cover different (but overlapping) portions of California, USA. The entire exercise would cover the terrain covered by two databases.

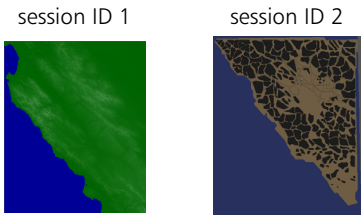


Figure 3-3. Using session IDs to load different terrain databases

If two different sessions use terrain databases that overlap, note the following possible sources of confusion:

- The front-ends in each session will see entities from the other session that are simulated in the common portions of the terrain. However, the front-end for one session will have no ability to control the entities that are simulated by the other session.
- If you save the scenario in one of these sessions, it only saves the entities simulated in the session. It does not save the entities from the other session, even though they may be visible in a front-end.
- If one of the sessions pauses or rewinds, the front-end in the other session will see the entities in the overlapping region pause or return to their origin.

For more information, please see "Specifying a Session ID," on page 5-8 and "Managing a Front-end's Session Connection," on page 5-14.

3.2.4. Coordinating Multiple Front-ends

Figure 3-1 shows how a VR-Forces front-end sends messages to a back-end and displays the simulation messages sent out by the back-end. When you run VR-Forces with multiple front-ends, any of them can send control messages to the back-ends and all of them display the state updates. However, you would not want users at different front-ends to send conflicting or overlapping messages. For example, you would not want two different users to try to load the same scenario. Therefore, when you run multiple front-ends, bear the following points in mind:

- Only one front-end in each session can load a scenario or create a new one (they can all create the objects in the scenario).
- Any front-end can save a scenario. However, this could lead to multiple, unsynchronized versions of a scenario. It is probably best to reserve saving of scenarios to a specific front-end.
- Any object that you create in any of the front-ends is visible in the others and can be managed from them.
- You can close the scenario from any front-end.

VR-Forces provides some help managing multiple front-ends. For example, if one front-end loads a scenario, VR-Forces automatically asks the other front-ends to load the appropriate terrain. And if you close a scenario in one front-end, VR-Forces automatically prompts the other front-ends to close their terrains.

3.2.5. Working with Multiple Back-ends

As noted previously, you can distribute simulation of entities and objects among multiple back-ends. The available back-ends are listed in the Selected Simulation Engine toolbar (Figure 3-4). If you are running multiple back-ends, when you create entities and objects, you specify which back-end you want to create them on by selecting a back-end in the toolbar. (For more information, please see Section 2.2, "Selecting the Object to Create," in *VR-Forces Scenario Management Guide*.)

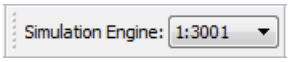


Figure 3-4. Selected Simulation Engine toolbar

When you run multiple back-ends, bear the following points in mind:

- When you save a scenario created using multiple back-ends, the object map file (Section 3.4, "The Object Map File," *VR-Forces Scenario Management Guide*) records each object and the site and application ID of the back-end on which it was simulated. When you want to replay that scenario, you should run a set of back-ends with the same site and application IDs. (If you do not, you can remap objects to the back-ends that you are running.)
- Multiple back-ends must all be started and ready when a scenario is loaded. There is no way for a required back-end to join an exercise after it has started and catch up to the current state of the exercise.
- If, during an exercise, a back-end drops out of the exercise, there is no good way for it to rejoin the exercise. If you load the scenario being run, it will be at the beginning, not at the point that the rest of the participants had reached.
- If you want a back-end to join an exercise after it has started, you must explicitly load the terrain database using `-T` command-line option. The back-end will not receive any order of battle data, but it can be used to create new entities and tactical graphics going forward in the simulation.
- When you save a scenario that uses multiple back-ends, the final scenario files are composed of information generated by each back-end and sent to the front-end, where it is combined. Therefore, if any of the back-ends has dropped out of the exercise or is otherwise unavailable, the scenario data will be incomplete.

Remapping Back-ends

The previous section lists the issues you should consider when creating, saving, and loading scenarios that use multiple back-ends. If you choose to run a scenario that requires multiple back-ends and some of the back-ends are not available, VR-Forces lets you remap objects from the missing back-ends to the available back-ends. You can remap back-ends automatically or manually.

If a scenario is missing its object map (*.omp*) file, you can remap the objects and create a new object map file. By default, in combined mode all the entities are remapped to the combined mode back-end. By default, in separate mode the remapping dialog box lists the missing back-end as "UNKNOWN".

You might also want to manually remap back-ends to load balance a scenario. For more information, please see "Load Balancing a Scenario," on page 1-18, in *VR-Forces Scenario Management Guide*.

3.3. Scenarios

Whenever you use VR-Forces, you work in the context of a scenario. A scenario consists of everything that needs to be specified to run a VR-Forces simulation. Each scenario has a scenario file (*.scn*) and the following supporting files:

- An order of battle file (*.oob*).
- A plan file (*.pln*).
- A global plan file (*.gpl*).
- An object map file (*.omp*).
- A terrain database for the front-end and back-end.
- An overlay file (*.ovl*).
- A scripts file (*.spt*). This file contains the scenario scripted tasks, if any.
- A selection groups file (*.sgr*).
- A scenario "extras" file (*.xtr*). This file records hostility relationships. If you have installed the B-HAVE Module, it stores spawn pattern templates.
- A saved views file (*.osrx*). If a scenario has captured observer views, they get saved automatically. You can also specify a startup view.
- A settings file that records scenario-specific GUI settings (*.gui_settings*).

A terrain database does not have to be stored in the same directory as a scenario. It is referenced by the scenario. This lets you use a terrain database, which can be quite large, in more than one scenario, without needing to make multiple copies of it. For more information about terrain databases, please see "Terrain Databases," on page 3-30.

For details about creating scenarios, please see *VR-Forces Scenario Management Guide*. For details about scenario files, please see Chapter 3, *Scenario Files*, in *VR-Forces Configuration Guide*.

3.4. Objects

All entities and tactical graphics are objects. The procedures for selecting, viewing, and organizing objects are fairly similar. However since their purposes and use are quite different, for the most part we discuss entities and tactical graphics separately.

3.4.1. The Object Parameter Database

VR-Forces creates objects based on parameter specifications in the object parameter database. In the object parameter database (OPD), object types are identified by an 8-field enumeration based on the DIS Enumeration Document (*Enumeration and Bit Encoded Values for Use with Protocols for DIS Applications*). (The DIS enumerations have also been incorporated into the HLA RPR FOM.) In this manual, when we refer to entity or force type enumerations, we are referring to this set of enumerations. For information about the object parameter database and how to edit it, please see *VR-Forces Configuration Guide*.

The objects in the OPD are organized into simulation model sets (SMS). An SMS contains all of the objects and supporting configuration files needed to populate a scenario. VR-Forces provides an SMS for entity-level simulation, *EntityLevel.sms*, and one for aggregate-level simulation, *AggregateLevel.sms*.

3.4.2. Local Objects and Remote Objects

From the point of view of a particular VR-Forces back-end, a local object is any object that is simulated in that back-end. In that same back-end, a remote object is an object that is received over the network from an external application, such as another VR-Forces simulation engine, or a non-VR-Forces simulation. A given VR-Forces application participating in a networked exercise may know about a mixture of local and remote objects.

There are two categories of remote objects: objects simulated by VR-Forces applications, and objects simulated by applications other than VR-Forces. When VR-Forces learns about a non-VR-Forces remote object, it knows only what is transmitted in the standard DIS or HLA message. By contrast, VR-Forces remote objects share additional state information over the network, including a string name, an echelon ID (if the entity is organized), and a string label.



The VR-Forces front-end allows you to manage all the objects in a VR-Forces session regardless of which back-end is simulating them. Therefore, in the interest of simplicity, we refer to entities or objects simulated by a VR-Forces back-end, whether local or remote, simply as entities or objects and we refer to entities simulated by applications other than VR-Forces as remote entities or objects.

You cannot control remote objects from VR-Forces. However, you can attach VR-Forces components such as sensors and radios to remote objects and you can write plans that use these components. For more information, please see Section 11.9, "Writing Plans for Remote Entities," in *VR-Forces Scenario Management Guide*, and Section 10.12, "Configuring VR-Forces Components on Remote Entities," in *VR-Forces Configuration Guide*.

3.5. Entity-Level Modeling and Aggregate-Level Modeling

VR-Forces supports two different approaches to modeling simulation objects – entity-level simulation and aggregate-level simulation.

In entity-level simulations, VR-Forces focuses on the interaction among individual entities, such as ships, tanks, automobiles, airplanes, and people. When you create an entity, you are usually creating one tank, one ship, or one human, and so on. Combat takes place between entities based on their weapon systems and their ability to withstand munitions.

In an entity-level simulation, you can combine individual entities into higher echelon units (aggregates), such as platoons and companies, but when they engage in combat, they do so by modeling the interactions of their members at the individual entity level.

Aggregate-level simulations focus on the high level movement and interaction of large numbers of personnel and equipment. For the most part, they do not model individual entity interactions and combat. When you create a simulation object, you are usually creating a company, or a brigade, and so on. Aggregate-level combat is modeled based on the relative strengths and weaknesses of simulation objects in personnel, equipment, and location, not on the engagement of individual entities.

As in entity-level simulations, in an aggregate-level simulation, you can combine entities into higher echelon units.

The type of modeling used in a scenario depends on the systems that simulation objects use, which are configured in the simulation model set it uses. If you use *EntityLevel.sms* or an SMS similar to it, it uses entity-level modeling. If you use *AggregateLevel.sms*, or an SMS similar to it, it uses aggregate-level modeling. Therefore, when you create a scenario, you must consider which type of aggregate modeling you want to use and you must remember which type of aggregate modeling you are using when you populate the scenario.

Aggregate-level modeling only works with HLA Evolved and the extended aggregate FOM. (If you try to create a scenario with *AggregateLevel.sms* and you are using an unsupported protocol, VR-Forces displays an error message.) For a detailed discussion of the aggregate warfare model, please see Chapter 5, *How Aggregate-Level Modeling Works*, in *VR-Forces Scenario Management Guide*.



AggregateLevel.sms has some simulation objects that represent individual platforms. However, like the objects that represent hiearachical units, they use data-driven models that are different from those used by similar entities in *EntityLevel.sms*.

Although it is possible to create a scenario that uses both types of simulation model sets, the resulting behavior will be unpredictable, because entities that use aggrgate-level modeling cannot interact with entities that use entity-level modeling. Combining these SMSs is not recommended and not supported.

3.5.1. Entity and Aggregate Terminology

Because we frequently refer to entity-level simulation and aggregate-level simulation, and because both approaches to simulation support individual entities and aggregate units, there is some potential for confusion about what the terms entity and aggregate mean and how they are used.

An entity is a player in a scenario. An entity can be:

- A simulation object that represents one platform, such as a tank, a ship, or a human. If we need to refer specifically to this type of entity, we call it an individual entity or platform.
- A simulation object that represents multiple platforms, hierarchical units, or both, such as a company or brigade, but which cannot be decomposed into those platforms.
- A group of simulation objects that have been aggregated into a higher-level unit such as a platoon, or a company. The aggregate can function as a unit, that is you can give it tasks and write a plan for it at the aggregate unit level. However, the members of the unit retain their identity and can function independently of the aggregate.

For most VR-Forces procedures, it does not matter which type of entity you are working with and we use the term entity generically. For example, Set Heading sets an entity's heading whether it is a tank or a brigade. In other cases it is obvious which type of entity the procedure refers to and there is no need to qualify the term. For example, Raise Periscope applies to a submarine. You would not try to give this task to a tank or any kind of aggregate.

If a procedure or feature applies only to a certain type of entity, then we indicate that in the text and with icons in the margin. If a section only applies to entity-level simulations, it has an entity icon () in the margin. If a section only applies to aggregate-level simulations, it has an aggregate icon () in the margin. .

3.6. Entities

The players in a simulation are called entities. There are two broad classes of entities: single entities, such as a tank or a submarine, and aggregate entities, such as a platoon or a company. Since aggregates are entities, they have most of the characteristics of single entities as described in the following sections.

The concepts described in this section are generally applicable to entity-level modeling and aggregate-level modeling. For details about the differences between aggregate-level modeling and entity-level modeling, please see "Entity-Level Modeling and Aggregate-Level Modeling," on page 3-11.

Entities can act in and respond to their environment. They execute tasks and they have behaviors. Each entity is part of a force-level organizational unit and may be a member of subordinate organizational units.

The VR-Forces application comes with a set of entity models that you can instantiate in a simulation. Application developers can create additional entities with the VR-Forces toolkit. You can also create new entity types in the Entity Editor. For information about creating new entities, please see *VR-Forces Configuration Guide*.

3.6.1. How Entities are Identified

VR-Forces identifies entities by name, echelon ID, object ID, and label. Table 3-1 summarizes the characteristics of each type of identifier.

Table 3-1: Entity identifiers

Identifier	Characteristics
Name	◆ Must be unique. ◆ Persists from exercise to exercise. ◆ Set by user. ◆ Limits on the length of the character string.
Echelon ID	◆ Must be unique. ◆ Persists from exercise to exercise. ◆ Managed by VR-Forces. Changes if an entity is organized into an aggregate.
Object ID	◆ Unique. ◆ Does not persist from exercise to exercise. ◆ No user control.
Label	◆ Does not have to be unique. ◆ Persists from exercise to exercise. ◆ Controlled by user. ◆ No limit on character length.

Entity Names

A name is a text string that, by default, consists of the entity category, as defined in the object parameter database, and an integer that in combination with the category, makes the name unique. For example, if you create a new scenario and add two M1A2 tanks and two Apache helicopters, the default names are `M1A2 1`, `M1A2 2`, `USAtRW 1`, and `USAtRW 2`. You can edit an entity name to make it meaningful to you.

Entity names have the following restrictions on their length:

- Entities – 11 characters.
- Aggregates – 31 characters.



If you are using a character set that requires multiple ASCII characters to represent a visible character, entity and aggregate names will need to be correspondingly shorter.

Echelon IDs

An echelon ID is a text string that identifies the entity's place in the force hierarchy. For example, the hypothetical entities in the previous section would be part of Blue force, or Force 1. The echelon ID for `M1A2 1` would be `1 M1A2, 1 Force`. The syntax of echelon IDs and the details of entity organization are described in "How Entities are Organized," on page 3-15. You cannot edit an entity's echelon ID. The echelon ID changes if you change the force hierarchy by adding, removing, or editing aggregate units.

Object IDs

Each entity has a unique object ID (global object designator). The global object designator is equivalent to the HLA object name in HLA, and the DIS entity identifier in DIS. The object ID is used to identify the entity internally and for network communications.

Entity Labels

An entity label is a text string that you can use to identify entities without the restrictions of uniqueness or character length that affect the other identifiers. For example, if Company A and Company C each have a 1st Platoon, you could not give both of them the name 1st Platoon, because names must be unique. However, you could give each of them the label 1st Platoon.

You can also identify entities using extended labels – additional text strings that you can give descriptive names to. For details, please see "Using Extended Labels," on page 7-10.



In the VR-Forces GUI, entity labels refer to informational graphics displayed in the 3D views. The text string label described in this section is simply called a label in the symbol decorations displayed in Plan View mode. For more information about the different types of labels and information displays, please see "Displaying Entity Labels," on page 7-3.

3.6.2. How Entities are Organized

All entities simulated by VR-Forces exist in the context of a hierarchy. At the top-most level, entities are grouped according to force ID (friendly, opposing, neutral, and so on). At lower levels, entities are grouped according to military organization, in which an entity has a designation in the hierarchy, such as 2nd Platoon, B Company. At each echelon level in the organization, an entity has a single superior entity. Aggregate entities contain one or more subordinate entities. The hierarchy's lowest level contains an individual entity.

Each entity has an echelon ID in the following format:

```
designator [category] [echelon], designator [category] [echelon], ...,
  force_number Force
```

where:

designator is a number or letter that differentiates the entity from other members of the echelon, such as 1 (M1A1), 2 (M1A2), and so on.

category is an optional descriptor for the entity, such as M1A2.

echelon is an optional descriptor for the organizational unit, such as "Platoon".

force_number is the force the entity belongs to, based on the force type enumerations. The force is the highest level in the hierarchy.

When you create a new entity, it exists only within the default force. Therefore its echelon ID would be similar to:

```
1 M1A2, 1 Force
```

If an entity becomes part of an aggregate, the aggregate information becomes part of the echelon ID, and the designator may change, for example:

```
2 M1A2, 2 Plt, 1 Force
```

The M1A2 from the first example is still an M1A2, but it has a new echelon designator: number two in `2 Plt`. It is still in Force 1.

How Echelon IDs are Assigned

The category and echelon labels that make up an echelon ID, such as “M1A2”, “Plt”, and so on, are specified in the VR-Forces object parameter database and are associated with object-types. When you create an entity or aggregate of a certain type, the echelon ID category and echelon name associated with that type are assigned to the entity. You cannot change the labels interactively. Although you can change the labels by editing the object parameter database, please note the cautions associated with doing so.

Designators are assigned by VR-Forces. You have no direct control over them. When you create new entities, they are assigned designators sequentially, starting with one (1). If you aggregate a group of entities, the aggregate leader is assigned the designator 1, and the other entities in the aggregate are assigned successive designator numbers.

3.6.3. How Entities Communicate

VR-Forces includes a default communication model called simple-radio-comm-model. Using this simple communication model, radio messages always reach their intended destination (as long as the destination is on a reachable network), and messages are all passed instantly. The simple radio communication model has one parameter, which determines how the radio networks are set up. There are three connection-modes available:

- `aggregate` mode specifies that each aggregate has its own radio network. Messages sent from an entity can reach any entity that is in the same organization, at the same level (the entity’s peers in the organization), and can also reach the entity’s direct superior. All other entities are outside the network, and are unable to receive messages sent by the entity.
- `force` mode specifies that there is a radio network for each force in VR-Forces. Messages sent from an entity can reach any other entity of the same force.
- `all` mode specifies that all entities in VR-Forces are part of one large radio network. In this case, any entity can receive messages sent by any other entity.

For details about configuring radios, please see *VR-Forces Configuration Guide*.

External Communication Model

VR-Forces includes an external communication model that can communicate with an external communication effects server to determine how to deliver radio messages. For details, please see Chapter 11, *Using an External Comms Effects Server*, in *VR-Forces Configuration Guide*.

Custom Communication Models

You can create new communication models and install them in VR-Forces. This is useful if you want to define your own connectivity model, or write your own algorithms for determining if radio messages get delivered. For details, please see API documentation.

3.6.4. VR-Forces Uses a Component Architecture

VR-Forces entities use a component architecture similar to that used in many robotics applications. There are three basic types of components supported: sensors, controllers, and actuators. Components communicate with each other through data ports and port groups, which may provide data as simple as a single number (such as a throttle value) or more complex data such as a list of entities that have been detected. The port groups make it possible to multiplex and prioritize multiple data sources to a single data recipient. For example, the automotive actuator of ground vehicles can receive inputs from a collision-avoidance controller, move-to-point controller, and follow-route controller, but it does not want inputs from all of those controllers at once.

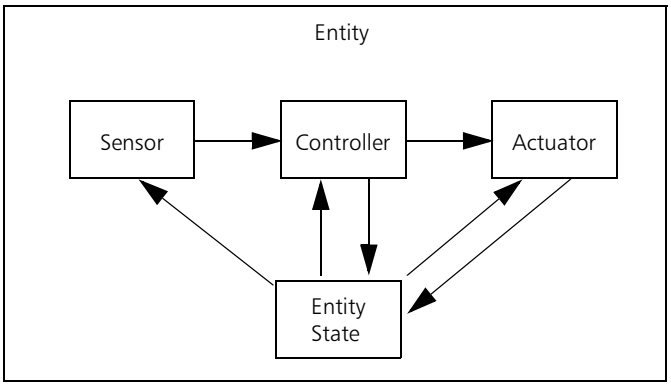


Figure 3-5. Entity component architecture

The component architecture supports any number of sensors, controllers, and actuators in a simulated entity. Components are specified in the object parameter database by using component descriptors. Component descriptors indicate the type of component that should be created. They can also provide parameter information to a component.

VR-Forces Configuration Guide discusses components in detail.

Sensors

Sensor components provide models of the simulated environment, which are used by controller components to make decisions and perform tasks. The simplest sensor might provide (simulated) ground truth, while more sophisticated ones could use complex models for sensing electromagnetic emissions, RADAR, or a cognitive model to simulate a soldier or crew member's perception of the simulated world. Sensor components can get information from:

- The virtual network.
- From a terrain database.
- By monitoring the state of the entity model.
- Many other potential sources.

The use of sensors is optional.

Controllers

Controller components typically provide control inputs to actuator components (although this is not required). Controllers can use information provided by sensors to perform their functions, but this is not a requirement either. For example, the controller that implements the various forms of the Wait task neither requires sensor information, nor provides control inputs to any actuators.

Actuators

Actuator components:

- Provide the physical model of the entity being simulated.
- Change the entity's state.
- Interact with other simulation objects (such as by sending messages).

Actuators can use control inputs provided by controller components as parameters to its model each simulation frame. Actuators may:

- Send messages on the radio system.
- Generate events on the virtual network, such as detonations.
- Modify the state of the entity simulated: position, velocity, damage, and so on.

Actuators can also use data directly from a sensor, when modelled or filtered data is preferred over simulated ground truth for a model that does not require a controller.

3.6.5. Entity Behaviors and Tasks

A behavior is something that an entity knows how to do on its own. Behaviors describe how an entity reacts to a specific condition or situation or how it performs a task. A task is something that you tell an entity to do. An entity does not innately know that it should move to a point. However, if you task an entity to move to a point, it knows how to locate the point and how to move to it.

Behaviors can override task assignments. For example, if an entity is moving along a route and detects another entity in its path, the collision-avoidance behavior overrides the movement task until the entity successfully avoids the obstacle and can return to moving along the route.

VR-Forces has behavioral models for the included entity types. These models simulate the behaviors you would expect from live participants and objects functioning in the real world. You can change the parameters for the behavior models. With the VR-Forces toolkit, you can change the models themselves and add additional models. For more information, please see Developers Guide.

To learn about changing behavior parameters, please see Chapter 10, *Configuring Object Behaviors*, in *VR-Forces Configuration Guide*.



No semi-automated forces (SAF) system, such as VR-Forces, can completely emulate the behavior of a complex real world entity. The behaviors provided by VR-Forces handle the lower level activities necessary to complete a task, and allow you to focus on the higher level activities involved in assigning tasks and preparing plans.

3.7. Aggregate Modeling

VR-Forces has two, quite different, ways to model entities – entity-level modeling and aggregate-level modeling. These modeling approaches affect how aggregate entities are modeled.

3.7.1. Aggregates in Entity-Level Scenarios

When you use entity-level modeling, VR-Forces models the behavior of individual (leaf-level) members of an aggregate. The members of an aggregate in an entity-level scenario can move individually through the scene, have their own plans, interact with other entities, and do all the things you would expect of an individual entity. When an aggregate engages in combat, the individual members of the aggregate fight the opposing forces using their weapons systems and survive or die as individuals.



Although it is theoretically possible for an entity-level scenario to have aggregates that cannot be disaggregated into individual vehicles and persons, *EntityLevel.sms* does not have any aggregates of this type and VR-Forces assumes that aggregates in entity-level scenarios can be disaggregated to individual entities.

Entity-level aggregates are simulated in either of two states:

- Disaggregated: Subordinates are simulated and published as individual entities (Figure 3-6). Aggregates in this state can be tasked and given plans, but usually pass responsibility for execution to their subordinates. For example, if you give a disaggregated aggregate a movement task, each member of the aggregate plots a path and follows it.
- Aggregated: Subordinates are not simulated separately. Tasks and plans are executed by the aggregate. For example, if you give an aggregated aggregate a movement task, the object representing the aggregate moves.

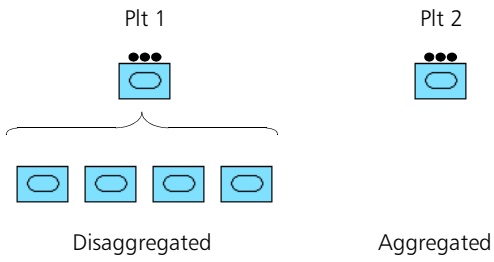


Figure 3-6. Disaggregated and aggregated entities



- The disaggregated and aggregated states are not the same as expanded and collapsed aggregates (which are explained in "Expanding and Collapsing Aggregates," on page 9-2.). Expanded and collapsed aggregates are simply different visual representations of the aggregate. Aggregated and disaggregated states are different simulation states.
- An aggregated aggregate in an entity-level scenario is similar to an aggregate in an aggregate-level scenario in that there is no representation of subordinates. However, an aggregated aggregate can disaggregate and model subordinates, whereas an aggregate in an aggregate-level scenario cannot do so. Furthermore, their underlying modeling is completely different.

An aggregate entity can support either of these states. At any given time, one subset of components and systems is enabled, and the other is disabled. At runtime, you can switch dynamically between these states.

You can configure VR-Forces so that aggregate state is determined automatically according to a set of rules, or to have aggregate state controlled manually.

The choice of whether or not to simulate aggregates in the aggregated or disaggregated state depends on your goals for the simulation. For example, if you need to support large numbers of entities, and do not want or need to simulate individual entity behavior, then you may want to simulate some or all of your aggregates in the aggregated state. Conversely, if you need to simulate the detailed behavior of subordinates, then you will want to simulate aggregates in the disaggregated state.

An aggregate in the aggregated state only simulates movement behavior. It does not model combat behavior. If you want an aggregate to model attrition, simulate it in the disaggregated state. An aggregate in the aggregated state keeps track of its current resource and attrition state. If an aggregate later transitions to a disaggregated state, this state information is preserved.

For more details about the differences between aggregated and disaggregated aggregates, please see Chapter 4, *Simulating Aggregates* in *VR-Forces Scenario Management Guide*.

How Aggregates Are Organized

The organization within a disaggregated aggregate is based on the echelon ID of each member entity. For example, in Table 3-2, four unique entities (identified by name) are assigned designators in `2 Plt`.

Table 3-2: Example of names and echelon IDs

Name	Echelon ID
Tank 1	1 M1A2, 2 Plt, 1 Force
Tank 2	2 M1A2, 2 Plt, 1 Force
Tank 3	3 M1A2, 2 Plt, 1 Force
Tank 4	4 M1A2, 2 Plt, 1 Force

Assignment of echelon IDs is handled automatically by VR-Forces when you create an aggregate.

Reorganizing Aggregates

Reorganization is the reassignment of echelon IDs to the members of an aggregate unit when a member entity gets destroyed. If reorganization is enabled, when a member of an aggregate unit gets destroyed, the destroyed entity is assigned the highest numerical ID in the unit, and the surviving members' echelon IDs are moved down numerically by one. For example, assume a platoon as shown in the leftmost table in Figure 3-7. If `2 M1A2` (currently the entity named Tank 2) gets destroyed, reorganization would result in the new echelon ID assignments illustrated in the rightmost table. The former `3 M1A2` is now `2 M1A2`, former `4 M1A2` is now `3 M1A2`, and the former `2 M1A2`, is now `4 M1A2`. Their echelon IDs have changed. Their names and object IDs have not.

Echelon ID	Name
1 M1A2	Tank 1
2 M1A2	Tank 2
3 M1A2	Tank 3
4 M1A2	Tank 4

Echelon ID	Name
1 M1A2	Tank 1
2 M1A2	Tank 3
3 M1A2	Tank 4
4 M1A2	Tank 2

Figure 3-7. Reorganization of Platoon 2

You can configure VR-Forces to reorganize aggregates automatically, or you can reorganize manually. Manual reorganization does not require you to reassign each designator; you simply order the aggregate to reorganize with the Reorganize command.



Reorganization only applies to disaggregated aggregates.

How Aggregates Move

When an aggregated aggregate entity receives a movement command, VR-Forces calculates the path it must follow. Then it moves to the location similarly to any individual entity. The entity icon is placed at the center of the bounding box of the area that would be encompassed by the aggregate if it was disaggregated.

When a disaggregated aggregate entity receives a movement command, VR-Forces calculates the path the aggregate leader must follow for movement. Then it calculates parallel paths (taking into account the formation) for each individual member of the aggregate. The individual entities are then responsible for following these paths until the aggregate completes the movement task.

Closing Formation

VR-Forces can close up holes in formations that are created when an aggregate member is removed from a formation. When an aggregate is ordered to form up, it assigns formation positions to each of its member entities. Each position in the formation has an internal ID. If at some point one or more of the aggregate's members are destroyed, removed, or independently tasked, the aggregate reassigns the formation IDs for its list of member entities. Member entities which are removed, destroyed, or independently tasked, are not given formation IDs. The active entities are moved up in the formation to fill the vacancies. Assignment to a different formation position has no effect on echelon designator.

The process of closing formation is internal to an aggregate. Formation position is not reported as part of entity information. The only way to determine an entity's formation position is to observe the formation as the aggregate moves.

Closing formation is an inherent feature of aggregates in the VR-Forces application. You do not need to do anything to put it into effect and you cannot turn it off.

Closing Formation Versus Reorganization

Closing the formation of the entities in an aggregate unit is not the same thing as reorganization. Reorganization changes the echelon designators of its members. Closing formation does not change the echelon designators of aggregate members, or the tasks an entity is assigned. It only affects the relative positioning of members within a formation.

How Aggregates Carry Out Tasks

Like other entities, each aggregate has a plan and you can assign independent tasks to aggregates (plans and tasks are discussed later in this chapter). When a disaggregated aggregate is given a task, it sends radio messages to its subordinates directing them to carry out their role in the task, for example, getting into formation and moving to a waypoint.

If an aggregate is in the aggregated state, it does not send task messages to subordinates. However, if while carrying out its task the entity disaggregates, it sends appropriate task messages to its subordinates to continue with the task execution. For example, suppose an aggregate is moving to a point and its path to the point passes through a disaggregation area. When it enters the disaggregation area, it would disaggregate, calculate routes for its subordinates, and send them Move Along Route tasks. When the aggregate exits the disaggregation area, it aggregates again and the tasks for its subordinates are abandoned.



Disaggregation areas are special control objects. If automatic aggregation and disaggregation is enabled, when an aggregate enters a disaggregation area, it automatically disaggregates. When it leaves the area it aggregates. For more information, please see Section 4.6.3, "Using Disaggregation Areas," in *VR-Forces Scenario Management Guide*.

The precise way in which an aggregate sends out messages to its subordinates is part of the modeling of the aggregate that is built into the VR-Forces application.

3.7.2. Creating Higher Echelon Aggregates in Aggregate-Level Scenarios

In aggregate-level scenarios, you can combine individual entities and aggregates to create higher echelon aggregates just as you can in entity-level scenarios. However, unlike in entity-level scenarios, you cannot switch these aggregates between an aggregated and disaggregated state. They are always disaggregated.

3.8. Tactical Graphics

VR-Forces lets you add tactical graphics to scenarios. Tactical graphics are objects that represent points, lines, areas, and information overlaid on the terrain. They are automatically assigned to overlays. (You can configure the default overlay assignments and can change the overlay at or after creation.) Entities have knowledge of some types of tactical graphics, which allows tasks and plans to take them into account. VR-Forces supports the following types of tactical graphics:

- Control objects. Entities are aware of control objects and can respond to them. They avoid obstacles, can move to waypoints, follow routes, and so on.
  - Waypoint. A specific X, Y, Z coordinate.
  - Phase line. Infinitely long line defined by two points.
  - Route. A sequence of points in three-dimensional space.
  - Area. A polygonal area. Areas are considered to have volume, so that air and subsurface entities can interact with them.
  - Obstacle. A polygonal area on the terrain that entities try to avoid if obstacle avoidance is enabled.
  - Disaggregation area. An area in which an aggregated aggregate automatically disaggregates if automatic disaggregation is enabled.
  - Terrain Page-in Area. An area in which streaming or paged terrain is given priority for paging in.
- Terrain Profile Line. A line that lets you examine the profile of the terrain along the line.
- Intervisibility Line. A line that shows line-of-sight between two points.
- Lines. Generic lines and lines with particular graphics, such as main axis and fortified line. Entities treat most lines as routes.
- Points. Various military symbology, such as point of contact and decision point. Entities treat most points as waypoints.
- Areas. Minefields and other areal objects. These are not treated like area control objects.
- Text.
- Symbols. Entity symbology that does not represent simulated entities.
- Circles, ellipses, and boxes.
- Combat engineering objects. Combat engineering objects are lines and areas that can impede the progress of entities and cause damage to them. Combat engineering objects are supported only in *AggregateLevel.sms*. For information about how the work, please see Section 5.2, "Combat Engineering Objects," in *VR-Forces Scenario Management Guide*.

Tactical graphics have names, which like entity names, must be unique. New objects are assigned default names in the format `Point 1`, `Point 2`, `Route 1`, `Route 2`, and so on. You can specify an alternative name when you create the object. A graphical object's name can be up to 255 characters long.

Tactical graphics can have labels and extended labels. Like entity labels, they do not have to be unique and do not have a length restriction.

Tactical graphics are assigned to a force. Waypoints are displayed using their force color. To determine the force of other tactical graphics, examine the Force attribute in the Environmental State page of the graphic's Information dialog box. For more information, please see "Displaying Information About an Object," on page 6-15.

The following sections provide more information about the basic control objects. For information about the other tactical graphics types mentioned in this section, please see:

- Chapter 14, *Intervisibility*.
- "Terrain Profiles," on page 13-2.
- Section 14.4, "Manually Paging-In Terrain," in *VR-Forces Configuration Guide*.
- Section 4.6.3, "Using Disaggregation Areas," in *VR-Forces Scenario Management Guide*.
- Chapter 2, *Creating and Placing Objects*, in *VR-Forces Scenario Management Guide*.
- Chapter 6, *Overlays and Tactical Graphics*, in *VR-Forces Scenario Management Guide*.

3.8.1. Points

A point identifies an X, Y, Z coordinate in the terrain. You can command an entity to travel to a point and to travel back and forth between points. Figure 3-8 illustrates waypoints in the 3D and 2D views. In 3D, the bottom of the symbol points to the location of the waypoint. In 2D, it is centered. You can place several different kinds of points on overlays.

A point can have an altitude value. This allows you to send aircraft to waypoints or have them patrol between points.

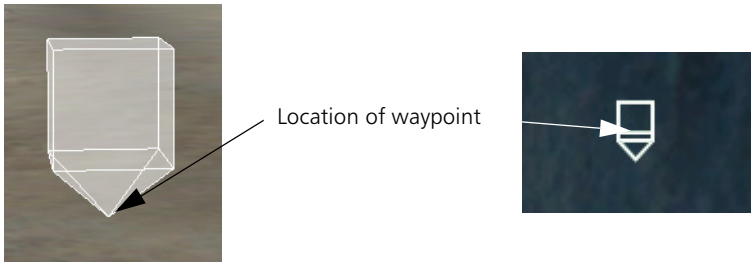


Figure 3-8. Waypoint symbol

3.8.2. Phase Lines

A phase line is defined by two points, but is considered to extend indefinitely in either direction and have infinite height. A phase line is used to test the location of an entity, for example: Is entity A to the left of phase line B? The left side of the line is determined relative to the direction in which the line is drawn. A phase line's label is placed at the beginning of the line. Therefore, you can always determine which side of a line is the left side. Figure 3-9 illustrates two phase lines. The arrows are placed to the left of the line and indicate the direction of the line.

Entities cannot move along phase lines, so do not create a phase line if you want a two point route.



Figure 3-9. Phase line direction

3.8.3. Routes

A line is a sequence of points in three-dimensional space. Like phase lines, lines have a direction. A line's name is displayed at its beginning.

Entities can move along lines from one end to the other, and can patrol back and forth along them.

Do not create a two-point route if you want a phase line. VR-Forces cannot determine whether an entity is to the left or right of a route.

3.8.4. Areas

An area defines a polygonal area on the terrain. (This is not a terrain polygon, it is just a graphic.) An area is used to test the location of an entity, for example: Is entity A in area B?

3.8.5. Obstacles

Obstacles represent areas of the terrain that would cause an entity problems, such as a ditch or minefield. In entity-level scenarios, if obstacle avoidance is enabled, if a ground entity encounters an obstacle in its path, it tries to drive around the obstacle.

Aggregate-level scenarios support combat engineering objects and contamination areas. These objects can defend, damage, or impede entities.

3.9. Tasks

VR-Forces entities can execute a variety of movement and tactical tasks. These tasks can be assigned in plans or through the Task menu. Tasks assigned outside of a plan are called independent tasks.

Some tasks are mutually exclusive with other tasks, for example, an entity cannot execute two different movement tasks at the same time. Other tasks, such as Send Text Message, can run concurrently with movement tasks. When you task an entity, if the new task is mutually exclusive with the current task, the entity immediately stops the current task and begins the new one. If the new task is not mutually exclusive with the current task, the entity executes the new task and continues with the previous task. For information about how to configure mutually exclusive tasks, please see Section 10.8, "Configuring Task Execution Rules," in *VR-Forces Configuration Guide*.

If an entity has a plan, when you give it an independent task, it abandons the rest of the plan.

Independent tasks are useful for assigning tasks to entities without the need to edit a plan. You can assign them interactively, so simulation participants can respond immediately to events in a simulation by assigning new tasks to an entity as they might have to do in the field. Tasks are described in detail in Chapter 7, *Assigning Tasks*, in *VR-Forces Scenario Management Guide*.

Many tasks are implemented as part of the VR-Forces code (C++ tasks) and cannot be added to or changed by end-users. Other tasks, called scripted tasks, are implemented using Lua scripts. You can edit scripted tasks and write new ones. For details, please see Section 16, "Creating Scripted Tasks," in *VR-Forces Scenario Management Guide*.

VR-Forces has a special type of scripted task, called a reactive task, that does not get explicitly assigned to entities. Reactive tasks monitor the simulation and run only if certain conditions are met. For information about reactive tasks, please see Section 7.3, "Reactive Tasks," in *VR-Forces Scenario Management Guide*.

A third type of scripted task is the background process. Background processes run continuously to handle certain types of ongoing aggregate-level activities, such as consuming or receiving supplies, calculating a unit's footprint, and automatically attacking opposing forces. For information about background processes, please see Section 17.11, "Background Processes," in *VR-Forces Scenario Management Guide*.

3.10. Plans

VR-Forces has two kinds of plans, global plans and entity plans. A plan can assign tasks and set data requests to an entity. Plans can also contain commands that create or delete entities and objects. The commands in a plan can be conditional, that is, they will be executed only if a condition is satisfied.

Global plans are created and executed independently of any particular entity. Commands sent to entities from a global plan affect an entity as if it received an independent task through the Task menu — they interrupt the current task, if any, and cause the entity to abandon its plan, if it has one.

An entity plan is a set of statements that order an entity to complete a sequence of tasks, change its state, or both. The entity carries out the tasks in order unless it is interrupted by an independent task or global command. When you open a scenario, it is in pause mode (unless you start it with the `-l` command line option). When you start the simulation, the plans start executing.

An entity owns its plan. Changing its name, echelon relationship, or force does not change its plan.

If an entity does not have a plan, it does not mean that the entity will have nothing to do. An entity may be assigned tasks by its superior in the force hierarchy. It can also be assigned tasks interactively from the front-end. If you want an entity to take planned actions independently of those assigned by its superior, you need to add statements to its plan.

You can view an entity's plan at any time in the Plan window. The task that an entity is currently executing is highlighted. A simple plan might look like the following:

```
Move-To Waypoint:"red point"
Wait-Duration Seconds-ToWait:900
Move-To Waypoint:"white point"
Set Engagement-Rules="Fire-at-will"
```


You can edit plans while a simulation is running. However, when you change an entity plan during a simulation, the entity starts executing the plan from the beginning, which may cause its activities to be out of synchronization with the other entities in the simulation.

Plans are discussed in detail in Chapter 11, *Writing Plans*, in *VR-Forces Scenario Management Guide*.

3.11. Set Data Requests

Set data requests affect the state or behavior of an entity, such as its speed or posture. They take effect immediately. Assignment of a set data request does not affect the task that an entity is executing. By contrast, if you assign an entity a task, it stops its current task and begins the new task. You can send set data requests from a plan or from the Set menu. The various set data requests are described in detail in Chapter 10, *Setting Entity State*, in *VR-Forces Scenario Management Guide*.

3.12. Terrain Databases

Each simulation takes place within the extents of a terrain database. A terrain database provides three-dimensional information about a geographic area and provides the environment within which simulated entities function. With the information in a terrain database, VR-Forces calculates height, intervisibility, and other three-dimensional information.

VR-Forces allows you to load terrain data in the following basic ways:

- Compose a terrain from elevation, imagery, and feature data.
- Open an MTF or MTD file.
- Stream terrain from local files or a terrain server such as VR-TheWorld Online.
- Open a terrain database in a supported format, such as FLT, DTED, or GDB.

The recommended work flow is:

1. Compose a terrain in the VR-Forces front-end and save it as an MTF file.
2. Create a scenario and choose the MTF file as the scenario database.

The rest of this section explains the terrain loading options. The decision making process for choosing a terrain when you create a scenario is described in Chapter 1, *Creating and Running Scenarios*, in *VR-Forces Scenario Management Guide*.



An MTD file is a descriptive file created by the MÄK Terrain Database Tool (TDB Tool). Some legacy terrains use MTD files, but the TDB Tool is no longer a recommended tool for creating terrains for VR-Forces.

3.12.1. Composing Terrains (and Creating MTF Files)

VR-Forces lets you compose terrains from a variety of supported data formats. You can combine elevation data (terrain patches), imagery, and feature data to create terrains.

VR-Forces offers the following basic methods for creating terrains:

- Compose a terrain by loading individual elevation, image, and feature data files. This method is best for relatively small terrains that do not require paging. It supports all of the coordinate systems that VR-Forces supports. For more information, please see Section 13.1, "Creating a Composed Terrain," in *VR-Forces Configuration Guide*.
- Specify a set of files to load using a *.earth* file. Use this method to stream terrain data from local or remote sources. This method is best for terrains with large amounts of elevation or image data, which would benefit from paging. SpeedTrees do not work with this terrain format. For more information, please see Section 14.1, "Connecting to Terrain Servers," in *VR-Forces Configuration Guide*.

When you compose a terrain, you can save it as a MÄK terrain file (MTF), which is essentially a description of all of the data used to visualize the simulated 2D and 3D environment. Saving a terrain lets you automatically load all of the data that you loaded manually when you first created it.

You can save terrains in a cached format that loads terrain data more quickly than if the terrain is created from the source data. For details, please see Section 13.9, "Configuring File Caching," in *VR-Forces Configuration Guide*.

3.12.2. Supported File Formats

VR-Forces can import elevation, terrain, images, models, and feature data. Some types of files can contain more than one type of information. For example GeoTiff files can have elevation data or just have image data. The following sections list the supported formats.



In theory, VR-Forces can load any format supported by osgEarth if you use the earth file format to load them. However, VR-Forces does not include drivers for all of the formats supported by osgEarth. Furthermore, when using osgEarth, VR-Forces only supports geocentric terrains. If you are uncertain about support for a particular format, please contact support@mak.com.

Terrain Data

VR-Forces imports terrain data directly as terrain patches. It supports the following formats:

- MÄK terrain database (*.gdb*).
- MÄK encrypted data format (*.medf*).
- MÄK Stealth (6.2 and earlier) format (*.mtl*).
- MÄK terrain project or terrain descriptor file (*.mtp*, *.mtd*).
- OpenFlight (*.flt*).
- MetaFlight (*.mft*).
- CTDB 4, 7, 7L (*.c4b*, *.c7b*, *.c7l*).
- Shape terrain data (*.shp*).
- Terrapage (*.txp*).
- Collada (*.dae*).
- 3DS.
- OBJ.
- OSG.
- LWO.

Elevation Data

VR-Forces supports the following types of elevation data:

- DTED (*.dt0*, *.dt1*, *.dt2*, *.dted*).
- GeoTiff
- DEM
- Arc/Info Binary Data (*.adf*).
- Other formats supported by osgEarth.

Supported Image Formats

You can import images in the following formats:

- GeoTiff (*.tif*, *.tiff*).
- JPEG (*.jpg*, *.jpeg*).
- PNG.
- RGB.
- Windows bitmap (*.bmp*).
- GIF.
- ECW.
- Other formats supported by the osgEarth using the GDAL driver. For a list of formats supported by GDAL, go to http://www.gdal.org/ and select the Supported Formats link. For more information about earth files, please see Section 14.1, "Connecting to Terrain Servers," and Appendix 15, *Editing Earth Files*, in *VR-Forces Configuration Guide*.

Supported Feature Data Formats

VR-Forces supports the following feature data formats:

- Shape files.
- S-57.
- OGR vector formats referenced in earth files.

3.12.3. Feature Layers and Props

A terrain can have objects such as buildings and trees embedded in it. These objects cannot be manipulated in this format. However, VR-Forces lets you extract these types of objects from the terrain and render them independently as props.

Props are terrain elements that you can manipulate as separate objects. You can add props to a terrain by extracting them from a terrain patch or from a feature layer.

A feature layer is a file that contains feature data such as point features, linear features, and areal features. You can extract point features as props. You can extract props from MÄK Terrain Format (GDB) files, shapefiles (SHP), or DFAD data (DFD).

For more information, please see Section 13.8, "Adding Props to a Terrain," in *VR-Forces Configuration Guide*.

Geometric Representation of Point Feature Data

There are differences between how the front-end and back-end process feature data. This is because the information needed to simulate entity interaction with features is different from the information needed to render the scene.

The front-end renders point features such as building or trees based on the data provided in the feature layer. As mentioned in the previous section, you can extract these objects as props, in which case they are represented as geometric objects in the terrain.

If you are streaming features using an earth file, you can configure the earth file to extrude buildings into 3D geometry. This capability is demonstrated in some of the insets in VR-TheWorld Online - MAK Earth, for example the Boston and Honolulu insets.

The back-end representation of point features depends on how they are used. Collision avoidance treats them as bounding volumes with infinite height. Line-of-sight calculations treat them as bounding volumes with a defined height. If you extract features as props, the back-end treats them as geometry where that is appropriate and as features otherwise.

The back-end does not recognize extruded geometry.

Feature Data Types Supported by the Back-end

The back-end supports the following types of feature data files:

- MÄK terrain database (*.gdb*).
- CTDB 4, 7, 7L (*.c4b*, *.c7b*, *.c7l*).
- Shape terrain data (*.shp*).
- S-57.
- WFS.
- VMAP.

The preferred format for adding features is shapefiles. You can extract feature data from GDB, DFD, VMAP, and CTDB into shapefiles with the vectorNetworkToShapeFiles utility. For details, please see Section H.7, "Extracting Feature Data from GDB to Shapefiles," in *VR-Forces Configuration Guide*. You can extract feature data from S-57 files into shapefiles with the ogr2ogr tool. For details, please see Section H.8, "Converting S-57 Feature Data to Shapefiles," in *VR-Forces Configuration Guide*.

3.12.4. Coordinate Systems

VR-Forces supports UTM, geocentric, flat earth, and Lambert Conformal Conic coordinate systems. There is no presumption that a particular database format, such as OpenFlight or DTED, will use a particular coordinate system.

The UTM Coordinate System

VR-Forces includes several databases that use Universal Transverse Mercator (UTM) coordinates. UTM databases are well-suited to two-dimensional views and simulations on a relatively small geographic area. However, their accuracy decreases as you move away from the origin, so they are not well-suited for exercises that need to simulate over areas greater than one UTM zone.

There are several types of UTM systems. In general, the earth is divided into 60 zones generally 6 degrees wide in longitude. These zones may be further subdivided into quadrangles by using letter designations from south to north. This complicates references to UTM. For example, standard offset could be used with or without the quadrangles (which affects the northing portion of the coordinate), non-offset does not use the quadrangles, and Milgrid requires the quadrangles.

More specifically, UTM coordinates define two dimensional, horizontal, positions. UTM zone numbers designate 6 degree longitudinal strips extending from 80 degrees South latitude to 84 degrees North latitude. UTM zone characters designate eight degree zones extending north and south from the equator. There are special UTM zones between 0 degrees and 36 degrees longitude above 72 degrees latitude and a special zone 32 between 56 degrees and 64 degrees north latitude.

VR-Forces uses the following definition:

A non-cartesian coordinate system in which the X, Y, and Z correspond to easting (nearly east), northing (nearly north), and height above an ellipsoid that approximates the surface of the earth. When in UTM mode (the default), VR-Forces uses an offset UTM coordinate system, one in which coordinates are expressed with respect to an origin located at the latitude and longitude specified in a configuration file. Typically, this will be the location of the origin of your terrain database.

Geocentric Terrain Databases

To support large play areas, simulations can use geocentric coordinates. In the geocentric coordinate system, the origin is at the center of the earth. The positive X-axis passes through the prime meridian at the equator; the positive Y-axis passes through 90 degrees east longitude at the equator; and the positive Z-axis passes though the North Pole.

3.12.5. Loading Supported Databases

MTF files reference elevation, image, and feature files that VR-Forces processes and loads when you load a scenario. However, you can also load terrain databases directly into VR-Forces. They will be converted to the VR-Forces internal format using default values. This process is less efficient than loading MTF terrains, may take more time, and must be repeated each time you load the terrain as part of a scenario or in the front-end.

When you create a scenario you can choose a terrain in any supported format. If you want to open a terrain using the **File** → **Open Terrain** menu option, you can only choose MTF files. To load non-MTF terrains out of the context of creating a scenario, you must load it as a terrain patch. For details, please see *VR-Forces Configuration Guide*.

If you load a terrain directly, you can still save it to MTF or GDB.

3.12.6. Using Large Terrain Databases

VR-Forces can support large databases, where large might mean a high polygon count, a large geographic area, or both. VR-Forces supports two methods for dealing with large databases – terrain paging and streaming terrain. Both approaches have a common goal – providing access to large amounts of data without loading all of that data into memory at any one time.

VR-Forces cannot arbitrarily page or stream terrains. It can page terrains that have been designed for paging and can stream terrains that are accessed through a terrain server, such as VR-TheWorld Server. For details about support for paged terrains, please see Section 14.2, "Loading MetaFlight Terrains," in *VR-Forces Configuration Guide*. For details about connecting to terrain servers, please see Section 14.1, "Connecting to Terrain Servers" and Section B.2, "Creating a Terrain with an Earth File," in *VR-Forces Configuration Guide*.



Even if you use terrain paging or streaming terrain, it is still possible that a large terrain could exceed the memory limitations of your operating system, particularly for 32-bit versions of VR-Forces.

Terrain Paging

Terrain paging breaks up a database into multiple cells that can be loaded independently. By comparison, a non-pageable terrain must be loaded in its entirety. When VR-Forces loads a paged terrain, the front-end loads the terrain at whatever level of detail is required by the zoom level. The simulation engine always loads terrain at the highest level of detail. Therefore, if you rapidly zoom into a section of the terrain in the front-end, there may be a delay in correlation between entities and the terrain until the front-end catches up to the proper level of detail.

The simulation engine does not load any terrain until it actually needs the terrain to support the objects in the scenario. For example, if you create a new scenario and specify a paged terrain, the simulation engine does not load any terrain until you create an entity or object that needs to be placed on the terrain. Then it only loads the terrain page required for the object.

When you run a scenario, as entities move, the simulation engine tries to predict which terrain pages it will need to support the entities and tries to page them in before the entities need them. If the total polygon count required to page in required terrain exceeds the maximum polygon count configured, VR-Forces pages out (removes) unused pages. The simulation engine pages out the least recently used data.



Predictive terrain paging is done in real time, not just when a scenario is running. For example, when you load a scenario, all required terrain is paged in and when you add objects during scenario creation, terrain is paged in.

You can also manually page in terrain using terrain page-in areas. For details, please see Section 14.4, "Manually Paging-In Terrain," in *VR-Forces Configuration Guide*.

Paging is done based on the following priorities:

- If an entity is moving, paging in terrain that it needs is a higher priority than predictive paging.
- Blocking terrain calls have a higher priority than non-blocking calls. (A blocking call is one that prevents the simulation from progressing until the call is completed. For more information, please see the **assertOnBlockingTerrainCalls** parameter in Table 2-1, in Section 2.2, "The vrfSim.mtl Configuration File," in *VR-Forces Configuration Guide*.)
- Paging out terrain to recover memory is a higher priority than paging in terrain. The need to page out is calculated based on the polygon count of the terrain, not on its memory usage.

VR-Forces can display information about paged terrains that can help you understand how your terrain is performing and why objects may be interacting with it as they do. For details, please see Section 14.5, "Displaying Information About Paged and Streaming Terrains," in *VR-Forces Configuration Guide*.

Terrain Streaming

The term terrain paging is used in reference to static, local terrains. Another way to load terrain data for a large area is to stream it from a terrain server. The idea is similar to terrain paging. Only the data you need at a particular location is loaded. The difference is that a terrain server can serve many computers at a time and normally holds much more data than a static local terrain. It can stream elevation data, imagery, map data, and feature data. It may have data at multiple levels of resolution for a particular portion of the terrain. VR-Forces supports connection configurations to terrain servers, including MÄK's VR-TheWorld Server.

3.13. Intervisibility (Line-of-Sight)

You can display intervisibility (line-of-sight or LOS) between points, within areas, and between a point and all entities in a defined area. Intervisibility displays can be transient or permanent. Permanent intervisibility displays are objects. You can resize them and move them around the terrain. Intervisibility objects can be tied to a particular entity.

You can display range and heading data for point-to-point intervisibility lines.

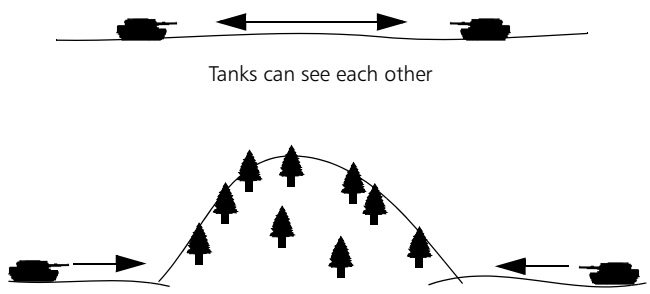


Figure 3-10. Intervisibility

You can configure parameters for intervisibility objects, such as the height above the terrain from which intervisibility is calculated. You can also change the colors used to display intervisibility. For procedures, please see Chapter 14, *Intervisibility*.

- By default, intervisibility between entities is calculated from the center of their bounding volumes. The VR-Forces target detection sensor uses an offset, which is often above the center of the entity. Unless you change the Origin LOS Elevation value in the Intervisibility Options dialog box to use the same offset as the target detection sensor, entities may shoot at targets for which they do not appear to have line-of-sight.
- Linear and areal features do not block line-of-sight. Only point features block LOS.

3.14. Representing and Managing Time in VR-Forces

VR-Forces maintains an exercise clock. The clock keeps track of elapsed simulation time, date, and time of day. In HLA federations, it may be affected by federation time. These concepts also have a relationship to wall-clock, or real time.

3.14.1. Simulation Time

Elapsed simulation time is the amount of time that has passed since the simulation started, and is expressed in days, hours, minutes, and seconds in the format 00:00:00:00.

Application developers can manipulate simulation time. Unless a local developer tells you otherwise, simulation time starts at 00:00:00:00.

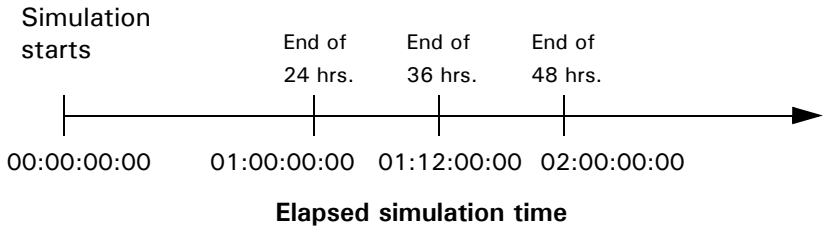


Figure 3-11. Simulation time

3.14.2. Time of Day

VR-Forces represents the day/night cycle with the Time of Day value. By default, it starts at 10:00 A.M in the time zone of the first entity created in a scenario, but can be set when you create a scenario. Time of day affects the simulation. For example, visual sensors are less able to detect objects at night than they are in the day.

Normally, time of day advances with simulation time, however you can change the time of day after the scenario is created, and even while the scenario is running. You can reset the time of day at any time though the environment settings, or though a plan with the Set Time of Day command. For details, please see Section 15.1, "Introduction to Environment Conditions," in *VR-Forces Scenario Management Guide*.

The display of time of day in the interface is 00:00 in hours and minutes. Time of day does not track number of days; when the time of day reaches midnight, is starts over at 00:00.

3.14.3. Exercise Clock Modes

The exercise clock supports the following modes:

- Variable-Frame Run-To-Complete mode advances simulation time by the amount of time passed since the last time the exercise clock was ticked. This mode is typical for distributed, interactive simulations. It does not provide repeatable results. Do not use this mode in time-managed HLA federations.
- Fixed-Frame Best-Effort mode advances simulation time by a fixed amount each frame, even if a frame takes longer than the fixed amount to compute. If the simulation takes less then the fixed frame time to compute, it waits until the frame time has elapsed before starting another.

  This mode is suitable for viewing or interactive use if your hardware can handle the frame rate. It is suitable for distributed use only in time-managed HLA federations.
- Fixed-Frame Run-To-Complete mode advances simulation time by a fixed amount each frame, even if a frame takes longer than the fixed amount to compute, just as in Fixed-Frame Best-Effort mode. However, if the simulation takes less then the fixed frame time to compute, it does not wait for the remainder of the frame time to elapse before starting the next frame.

  This mode is most useful for situations where you want a simulation to run with internal consistency and high fidelity, and want it to run to completion, but do not need to observe the simulation. So, for example, you might run a simulation overnight and view the results the following day. Fixed-Frame Run-To-Complete mode is not suited for interactive use. It is suitable for distributed use only in time-managed HLA federations.



If you want to change the default clock mode, you must use the correct syntax. For information about how to set the exercise clock mode, please see Chapter 3, *Scenario Files* in *VR-Forces Configuration Guide*.

Running a Simulation Faster than Real Time

VR-Forces supports simulation at rates faster than real-time by applying a multiplier to the tick time.



- Do not run your simulation at faster than real-time if you are interacting with other real-time simulations.
- Running a simulation faster than real-time can cause performance of entity models to degrade.

3.14.4. Using HLA Time Management

In HLA, use of time management allows the rate at which a federate proceeds to be controlled by another federate. Federates being controlled by other federates are said to be constrained. Federates that control other federates are time regulating. A federate can be both time regulating and constrained. It can also be unconstrained.

The main reasons why you might want to enable time management for VR-Forces are to run VR-Forces as part of a wider federation that is using time management or to synchronize multiple VR-Forces back-ends. Without time management, multiple back-ends can stay synchronized only if they all are able to maintain the same frame rate. However, if they cannot do so, they will no longer have a shared concept of simulation time and will become unsynchronized. By using time management services, multiple back-ends can be configured to run in lock-step with respect to simulation time, ensuring that events will always happen in the same order.



VR-Forces supports time management in the back-end only.

For details about enabling time management, please see "Configuring Time Management for HLA Exercises," on page 5-17.

VR-Forces Simulation Time Versus Federation Time

Time management synchronizes federates to federation time. Federation time is the common model of time for the federation as whole. It is the time used in time stamp ordered messages (TSO), exchanged by time-managed federates. During the lifetime of a federation, federation time always increases. (There is no notion of rewinding back to an earlier time as with VR-Forces simulation time.)

VR-Forces simulation time is not the same as HLA federation time. It is the elapsed time within a scenario (the time displayed in the Simulation Time toolbar). When you first start a simulation, federation time and simulation time may be similar. However, if you rewind a time-managed simulation, the discrepancy between the two time streams will increase. This should not have any effect on running and rewinding scenarios. As long as all the back-ends start from the same simulation state when you begin a simulation, they will be correctly synchronized (against federation time) as the simulation progresses.

3.15. Interactions

VR-Forces displays interactions, such as detonations, that it receives in interaction messages or PDUs. You cannot select or otherwise manipulate them. You can only view them.



4. VR-Forces Visualization Concepts

This chapter briefly describes the concepts underlying visualization of simulations in the VR-Forces window.

The Display Engine 4-2
- Window Types 4-2

The Observer 4-2
- Switching Between the 2D and 3D Projections 4-3
- Observer Modes 4-3
- Introduction to XR Mode 4-5

Introduction to VR-Forces Object Modeling 4-6
- Model Formats for 3D Entities, Objects, and Effects 4-8
- Model Sets 4-9
- Mapping Entity Types to Element Definitions 4-9

4.1. The Display Engine

The VR-Forces front-end has a component called a display engine. The display engine renders visual data and can display it in one or more windows, which can each have zero or more channels. The windows and channels being rendered at any particular time make up a display configuration. You can save display configurations and can load them to quickly recreate a particular configuration.

4.1.1. Window Types

VR-Forces supports the following types of windows:

- Full screen. A full screen window uses the entire area of a monitor. It does not have any menus, panels, or other GUI controls.
- Resizable. A resizable window can be moved, resized, maximized, and minimized. It does not have any GUI controls. An inset view is an example of a resizable window. (For information about inset views, please see "Inset Views," on page 8-7, in *VR-Forces Users Guide*.)
- Embedded. An embedded window is part of an application. The main display window in the VR-Forces front-end is an embedded window. The application provides a variety of controls that affect the view in the window. You could, if you wanted to, change VR-Forces's window to a different type, in which case it would no longer be embedded in the main application window.

A window can have zero or more channels. Each channel defines a viewable area on the screen. You must have at least one channel to see a rendered image in the window. Each channel can have its own observer.

4.2. The Observer

The observer, or eyepoint, is the location in the 3D or 2D environment from which you observe the scene. You can move the observer (navigate) through the scene. You can attach the observer to an entity or prop. When the observer is attached to an entity, it moves automatically with that entity. You can also control the observer from a remote application using view control messages.

You can have multiple observers in a VR-Forces session. They might be associated with different windows. For example you could have an observer for the main window and one for an inset window.

The observer is such a pervasive component of the VR-Forces front-end that it is not covered in just one section of this manual. For more information, please see:

- Chapter 10, *The Observer and Observer Modes*.
- Chapter 11, *Moving the Observer*.
- Chapter 12, *Attaching the Observer to Objects*.

4.2.1. Switching Between the 2D and 3D Projections

VR-Forces supports both 2D and 3D projections. You can switch between them simply by changing the observer mode from Stealth or Out-the-Window, to Plan View. (Changing observer modes is described in "Changing the Observer Mode," on page 10-2.) When you go from a 3D observer mode to 2D, VR-Forces calculates a zoom level so that the observer should appear as if it is roughly the same height above the terrain. When you go from a 2D observer mode to 3D, VR-Forces calculates an altitude to do the same thing, and keeps the 3D view pointing down (so the change in overall view stays fairly small).

4.2.2. Observer Modes

Observer modes let you configure VR-Forces settings that affect how the observer moves and what it sees. You can then quickly change from one group of settings to another by switching between observer modes. An observer mode specifies:

- A model set. A collection of models of a particular type, such as 3D models.
- A key map. A coordinated set of key mappings to movement functions.
- A projection (2D or 3D).
- Settings for entity graphics and observer movement.

VR-Forces includes the following observer modes:

- Stealth. Stealth observer mode is a 3D projection configured for situational awareness.
- Out-the-Window. Out-the-Window observer mode is configured for when you want to use VR-Forces as a desktop image generator. It hides many of the explanatory visual effects present in the Stealth observer mode.
- Plan View. The Plan View observer mode provides a 2D "big picture" view of the terrain.
- XR. XR, or exaggerated reality, observer mode is a 3D mode that helps you understand the relationships between entities and the terrain when the observer is too far away from entities to view them at realistic sizes. It allows you to exaggerate entity size and terrain.
- IR BlackHot (CameraFX). A view using the CameraFX IR BlackHot sensor.
- IR (CameraFX). A view using the CameraFX IR sensor.
- NVG (CameraFX). A view using the CameraFX NVG sensor.
- Camera (CameraFX). A view using the Visual Camera CameraFX sensor.

Table 4-1 lists the observer modes and default settings provided with VR-Forces.

Table 4-1: Observer modes

Observer Mode	Projection	Key Map
CameraFX modes	3D	Observer Frame
Stealth	3D	Observer Frame
XR	3D	Observer Frame
Plan View	2D	2D Level Observer Frame
The observer modes all support the three model sets: 3D models, 3D colorized models, and 2D icons.		

You can edit observer modes and add new ones. For details, please see "Creating and Editing Observer Modes," on page 10-2.

4.2.3. Introduction to XR Mode

The Stealth observer mode (3D projection and 3D model set) provides situational awareness at the entity or aggregate level. It is not ideal for getting the big picture of a simulation, because when you move the observer out to see large areas of terrain, the entity models, which are scaled to the terrain, are too small to see. The Plan View observer mode provides the big picture of a simulation, but does not provide situational awareness.

XR observer mode bridges the gap between the Stealth and Plan View observer modes. It is a 3D mode, so it provides situational awareness. However, it uses a special model set that makes models visible at greater distances than in Stealth observer mode. The 3D colorized model set shows entities as notional models using force coloring, which helps them stand out from the terrain. It also scales the model size to provided an exaggerated reality (hence XR). You can also scale the regular 3D model set if you want to. Figure 4-1 illustrates model scaling with 3D colorized models.

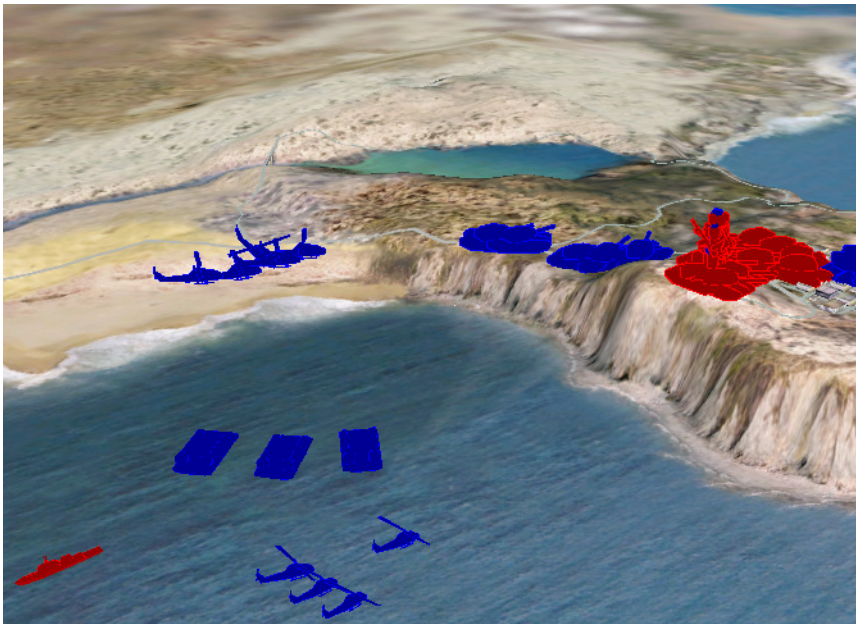


Figure 4-1. Model scaling and colorized models

For details about scaling models, please see "Exaggerating the Scale of 3D Entity Models," on page 7-16.

4.3. Introduction to VR-Forces Object Modeling

The objects that VR-Forces can display that are not part of the terrain, such as entities, tactical graphics, detonations, and cockpit displays, are generically called scene elements. Scene elements are defined in element definitions. When VR-Forces runs a scenario, it publishes information about the objects it is simulating. It may also receive object information over the network from other simulation participants. The objects are identified using entity type enumerations. VR-Forces maps these entity types to element definitions. (For more information about entity type enumerations and mapping them to element definitions, please see "Mapping Entity Types to Element Definitions," on page 4-9.)

An element definition is a set of visual definitions that tell VR-Forces how to render (display) the object. Visual definitions specify the 3D model, 2D icon, or graphic, and the visual effects that the object supports, such as trailing effects, height-above-terrain lines, track histories, and so on. A scene element can have up to three different element definitions – one for each model set (3D, 3D Colorized, 2D Icon). VR-Forces chooses the correct element definition to use based on the model set being used by the observer. (For more information about model sets, please see "Model Sets," on page 4-9.)

Element definitions are created and edited in the Visual Model Editors dialog box. In addition to letting you manage element definitions, this dialog box lets you create and manage model definitions and schemas. Model definitions specify values for some basic object parameters. Most importantly for most VR-Forces users, model definitions specify the 3D model to use for element definitions in the 3D Models model set. What this means is that if you want to add a new model to VR-Forces, you will specify the model file, such as an OpenFlight file, in a model definition and then specify that model definition in the element definition for the entity you want to add.

Schemas define the most basic scene elements, such as entity, SpeedTree, or line. Most VR-Forces users will never have to create or edit a schema. However, if you create a new model definition, you will need to make sure it uses the correct schema and that it includes all required parameters.

Figure 4-2 illustrates these relationships using an M1A2 entity. VR-Forces receives the entity type 1:1:225:1:1:3:-1 over the network (callout 1). In the Entity Type Mappings dialog box, this entity type is mapped to the M1A2 Abrams entity definition (2). (An entity definition is a specific type of element definition.) The M1A2 Abrams entity definition (in the Visual Model Editors dialog box) specifies all of the visual definitions for this entity (3). The Main Model visual definition has a **modeldefinition** parameter. The value for this parameter is TrackedM1A2DESERT (4). The TrackedM1A2DESERT model definition has a **filename** parameter that specifies *M1A2.medf* as the model to use for entities that use this model definition. The TrackedM1A2DESERT model definition has a **filename** parameter because it uses the Entity schema and the Entity schema requires a **filename** (5).

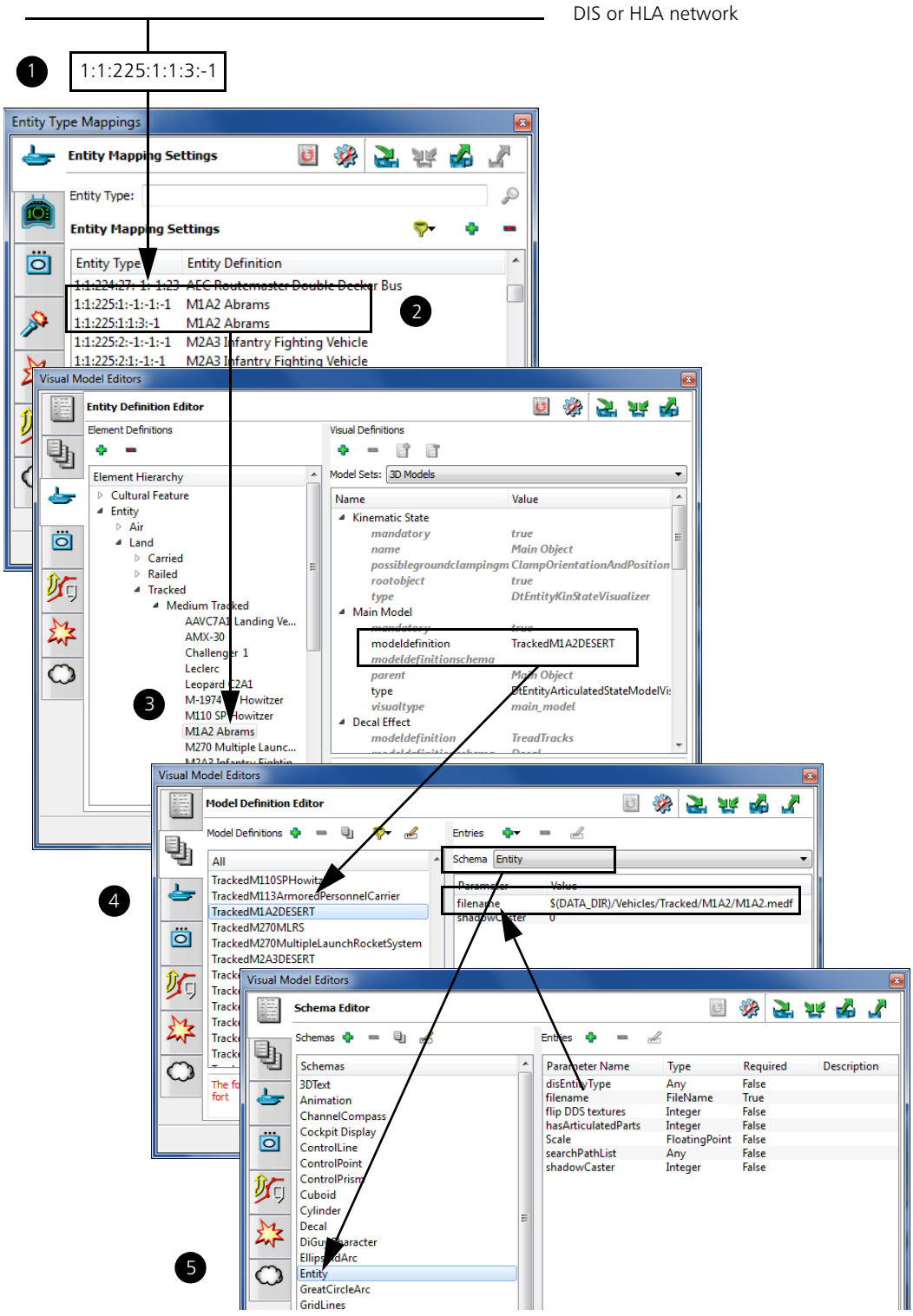


Figure 4-2. Element definitions and model mapping



When you add entities and other scene elements in the Entity Editor, it creates model definitions and handles all of the mappings described in the previous paragraphs and the illustration. Although *VR-Forces Configuration Guide* explains all of the details of the Visual Model Editors dialog box and Entity Type Mappings dialog box, using the Entity Editor to add entities is the preferred method.

4.3.1. Model Formats for 3D Entities, Objects, and Effects

VR-Forces includes 3D models for vehicles, lifeforms, environmental objects, and effects. The models are in a compressed, encrypted format (extension *.medf*).



VR-Forces includes a utility, the MEDF Tool, that lets you encrypt files in MEDF format. You cannot use it to decrypt encrypted files. For more information, please see Section 16.3, "Compressing Model Files," in *VR-Forces Configuration Guide*.

You can provide your own models if they are in one of the following supported formats:

- OpenFlight
- 3ds
- OBJ.
- COLLADA (*.dae*).
- OSG (*.osg*, *.osgb*, and *.ive*).
- NewTek LightWave (*.lwo* and *.lws*).
- MÄK GDB.

4.3.2. Model Sets

The model or icon that is used for an object is determined by its element definition. For convenience, element definitions are organized into model sets. For example, all of the element definitions that specify 3D models are in the 3D Models model set, and so on. As a result, when you choose the 3D Models model set for an observer mode, all of the entities use a 3D model. (The model set is an observer-specific setting.)

VR-Forces includes the following types of object models and icons:

- 3D Models. Realistic 3D models. This is the default model set for the Stealth observer mode.
- 3D Colorized Models. 3D colorized models are cartoon-like models that create an exaggerated notional view of the world. This is the default model set for XR observer mode.
- 2D Icons. 2D MIL-STD 2525B font-based icons. This is the default model set for the Plan View observer mode. You can also configure VR-Forces to use 2525a images or any arbitrary images for 2D icons.



The model sets are fixed. You cannot add model sets.

For information about changing model sets, please see "Changing the Model Set," on page 7-15.

4.3.3. Mapping Entity Types to Element Definitions

Entity type mapping is the process of associating an entity type enumeration for an entity or other scene element with an element definition.

DIS and HLA objects are specified by an entity type enumeration. An entity type enumeration is a list of seven components as specified by the DIS protocol and HLA RPR FOM. If none of the seven components contains a wildcard (-1) value, then the entity type refers to a specific entity. If some components contain a wildcard, then the entity type refers to a class of entities. A larger number of wildcards indicates a broader, more general class. For complete details about entity type enumerations, please see *Enumeration and Bit Encoded Values for Use with Protocols for DIS Applications* (DIS enumerations document). It is available at the IEEE World Wide Web site: http://shop.ieee.org/store/ and at the SISO web site: http://www.sisostds.org/.

When VR-Forces discovers an entity, it reads the enumeration and then displays the entity using the element definition that is mapped to the enumeration.



5. Starting VR-Forces

This chapter explains how to start and stop VR-Forces and load terrain databases.

Starting VR-Forces 5-3
- Starting VR-Forces from the VR-Forces Launcher 5-4
- Starting Independent VR-Forces Executables 5-8
- Specifying a Session ID 5-8
- VR-Forces Startup Tutorial 5-8

The VR-Forces Window 5-11
- Opening New Windows 5-13
- Printing the VR-Forces Window 5-13

Managing a Front-end's Session Connection 5-14
- Joining a Session 5-15
- Resigning from a Session 5-15
- Configuring Session Messages and Join at Startup 5-16

Configuring Time Management for HLA Exercises 5-17
- RTI Requirements for Time Management 5-18
- Enabling Time Management for the Back-end 5-18

Opening a Terrain Database 5-19
- Loading a Terrain Database at Startup 5-21
- Saving a Terrain Database in GDB Format 5-21
- Closing a Terrain 5-21

Managing VR-Forces Settings 5-22
- Synchronizing Settings Among VR-Forces Installations 5-24
- Global Settings and Observer-Specific Settings 5-24

Exiting VR-Forces 5-24

Configuring Simulation Connections 5-25

Opening the Simulation Connections Configuration Dialog Box..... 5-25
Adding a Simulation Connection 5-26
Editing a Simulation Connection 5-27
Simulation Connection Parameters 5-28
Deleting a Simulation Connection 5-30
Configuring Auto Connect 5-30
Displaying Connection Information..... 5-31
Managing Plug-ins..... 5-32
Loading Plug-ins 5-32
Adding a Plug-in 5-35
Specifying the DLLs for a Plug-in 5-36
Adding a Plug-in Configuration 5-37
Deleting a Plug-in Configuration 5-37
Deleting a Plug-in 5-38
Viewing a List of Loaded Plug-ins 5-38
The VR-Forces Log Files..... 5-39

5.1. Starting VR-Forces

In "The VR-Forces Program," on page 3-3, we describe the VR-Forces front-end and back-end applications. You can start front-ends and back-ends separately, which is called separate mode, or you can use the VR-Forces Launcher to start them together, which is called combined mode. The only real difference between the two startup modes is that in combined mode, when you shut down the front-end, the back-end automatically shuts down also.

By default, all front-ends and back-ends start with session ID 1. Front-ends automatically try to join a session. If no back-ends are running when a front-end starts, it does not join a session. However, if at a later time a back-end with the same session ID starts, the front-end will automatically join the session.



If you run VR-Forces for HLA, be sure that you have installed an RTI and configured your system to use it. For details, please see "Installing an RTI," on page 2-9.

For convenience you can create batch files or scripts to start VR-Forces. Please see Appendix B, *Command-line Options* for a list of command-line options.

5.1.1. Starting VR-Forces from the VR-Forces Launcher

You can run VR-Forces using DIS, HLA 1.3, HLA 1516, or HLA Evolved. For the HLA specifications, you can run using one of several different versions of the RPR FOM. With all these different possible startup configurations, particularly for HLA, it can be confusing to keep track of the options for starting VR-Forces. The VR-Forces Launcher simplifies the process of specifying the simulation standard to use when starting VR-Forces.

The VR-Forces Launcher lets you start in combined mode or separate mode.

To start VR-Forces using the VR-Forces Launcher:

1. On Windows, on the Start menu, choose **Programs** → **MAK Technologies** → **VR-Forces 4.3.1** → **VR-Forces** *configuration*, where *configuration* is:

   – VR-Forces GUI

   – VR-Forces Simulation Engine

   – VR-Forces GUI + Simulation Engine.

   To start the Launcher on Linux (or from the Windows command line), in a console window, enter the following command:

   ```
   vrfLauncher options
   ```

   For descriptions of the vrfLauncher command-line options, please see "Command-line Options for vrfLauncher," on page B-12. The basic startup configurations for vrfLauncher are:

   – Start vrfGui and vrfSim (combined mode):

   ```
   vrfLauncher
   ```

   – Start vrfGui only:

   ```
   vrfLauncher -F
   ```

   – Start vrfSim only:

   ```
   vrfLauncher -B
   ```

   – Start in combined mode and do not display the Simulation Connections Configuration dialog box:

   ```
   vrfLauncher --usePredefinedConnection DIS
   ```

If this is the first time that you are running VR-Forces, or if you have not specified a default startup connection, the Simulation Connections Configuration dialog box opens (Figure 5-1).

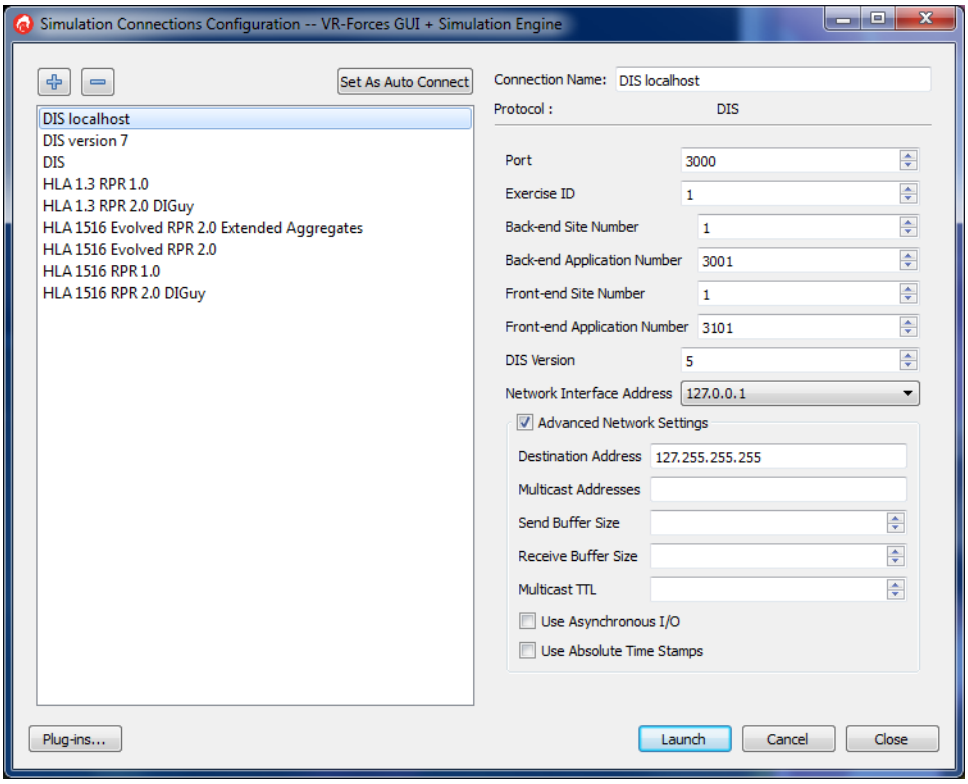


Figure 5-1. Simulation Connections Configuration dialog box (DIS)

2. In the Simulation Connections Configuration dialog box, select a connection configuration. (Optionally, create a new configuration. For details about creating and editing simulation connections, please see "Configuring Simulation Connections," on page 5-25.)
3. Optionally, edit the connection's parameters. If you change a parameter it is saved automatically.
4. Click Launch. The VR-Forces executables start. When the VR-Forces applications finish their startup process, the Scenario Startup dialog box opens (Figure 5-2). It lets you quickly create or load a scenario using common options.



Figure 5-2. Scenario Startup dialog box

5. To create a scenario or load a scenario, select one of the options and click OK. The options are as follows:
   - Create a new scenario using streaming terrain from VR-TheWorld Online. This option connects to the VR-TheWorld terrain server. It lets you choose map configurations that have different levels of resolution to suit the performance needs of your scenario. The Status field indicates whether or not VR-Forces is able to connect to the selected terrain server using the specified connection.
   - Create a new scenario on a local terrain. This option is the same as choosing **File** → **New Scenario**.
   - Load a scenario. This option is the same as choosing **File** → **Load Scenario**.
6. If you do not want any of the options, click Close. (This closes the Scenario Setup dialog box; it does not close VR-Forces.)

Enabling and Disabling the Scenario Startup Dialog Box

If you do not want VR-Forces to display the Scenario Startup dialog box every time it starts up, you can disable it. You can enable the dialog box at a later time.

To enable or disable the Scenario Startup dialog box:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the Session Options page (Figure 5-3).

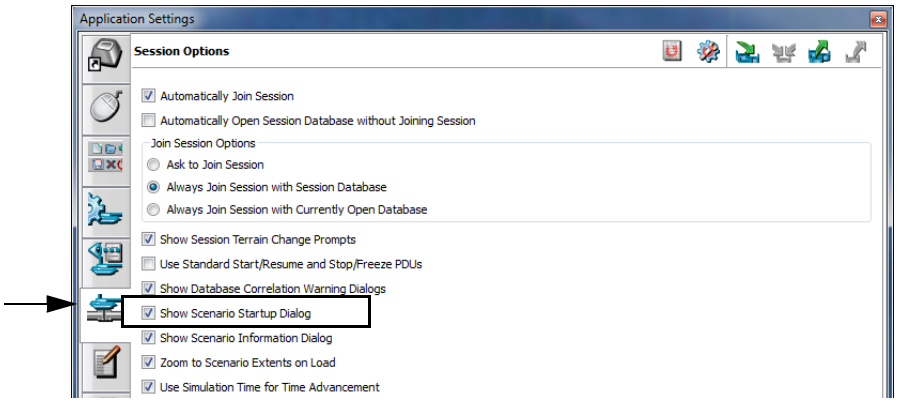


Figure 5-3. Session Options page

3. Select or clear the Show Scenario Startup Dialog check box.



You can also disable display of the Scenario Startup dialog box by selecting Do Not Show This Dialog At Startup on the dialog box itself.

5.1.2. Starting Independent VR-Forces Executables

To start independent VR-Forces executables:

1. Open a console window.
2. Change to the *./bin* directory.
3. Enter a command for each VR-Forces executable you want to run. You must specify the protocol to use. For vrfSim, the protocol is part of the executable name. For vrfGui, it is a command-line argument. Specify a unique site:application identifier for each executable, for example:

   ```
   vrfGui -s 1 -a 3000 {--dis | --hla13 | --hla1516 |
     --hla1516e} options
   vrfSimprotocol -s 1 -a 3001 options
   vrfSimprotocol -s 2 -a 3001 options
   ```

   where *protocol* is `DIS`, `HLA13`, `HLA1516`, or `HLA1516e`.

A VR-Forces window opens for each front-end you start and a console window opens for each back-end. For an example of running multiple front and back-ends, please see "VR-Forces Startup Tutorial," on page 5-8.

5.1.3. Specifying a Session ID

➤ To assign a session ID, use the `-i` command line option.

```
vrfGui -s 1 -a 3000 -i 25 {--dis | --hla13 | --hla1516 |
  --hla1516e} options]
vrfSimprotocol -s 2 -a 3002 -i 25
```

where *protocol* is `DIS`, `HLA13`, `HLA1516`, or `HLA1516e`.

The default session ID is 1. You cannot change the session ID during runtime.

5.1.4. VR-Forces Startup Tutorial

This section is a brief example of how to run multiple front and back-ends on multiple computers. Examples are provided for DIS and HLA. For both protocols, assume the configuration in Figure 5-4. This configuration has two front-ends and three back-ends. Each is on a different computer. Assume that they are all running in the same session.

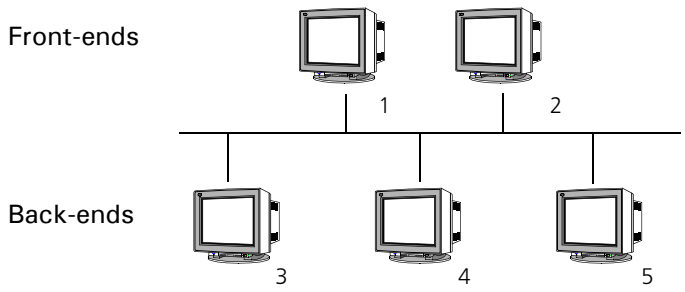


Figure 5-4. VR-Forces multicomputer configuration



As an alternative to starting individual front-ends and back-ends from the command line and specifying their protocol information in the command line, if you do not need to specify other command-line options, you can use vrfLauncher to start the front-ends and back-ends. Be sure to specify unique site and application numbers in vrfLauncher. For details about vrfLauncher command syntax, please see "Command-line Options for vrfLauncher," on page B-12.

DIS Configuration

In a DIS exercise, you typically specify a port so that the exercise traffic is isolated from other network traffic. Each front-end and back-end must have a unique site:application ID.

Run VR-Forces on each PC using the following commands:

- PC 1:

```
vrfGui -s 1 -a 3000 --dis -P 1234
```

- PC 2:

```
vrfGui -s 1 -a 3001 --dis -P 1234
```

- PC 3:

```
vrfSimDIS -s 1 -a 3002 -P 1234
```

- PC 4:

```
vrfSimDIS -s 1 -a 3003 -P 1234
```

- PC 5:

```
vrfSimDIS -s 1 -a 3004 -P 1234
```

HLA Configuration

In an HLA exercise, you could simply specify the site:application ID and use default values to specify the HLA configuration parameters. Specification of the site:application ID is identical to that in DIS. For this example, we use HLA 1.3 and specify a FOM Mapper other than the default (RPR FOM 1).

Run VR-Forces on each PC using the following commands:

- PC 1:

```
vrfGui -s 1 -a 3000 --hla -x VR-Link20017-1
  --rprFomVersion 2.0017 -F VR-Link20017-1.fed
```

- PC 2:

```
vrfGui -s 1 -a 3001 --hla -x VR-Link20017-1
  --rprFomVersion 2.0017 -F VR-Link20017-1.fed
```

- PC 3:

```
vrfSimHLA13 -s 1 -a 3002 -x VR-Link20017-1
  --rprFomVersion 2.0017 -F VR-Link20017-1.fed
```

- PC 4:

```
vrfSimHLA13 -s 1 -a 3003 -x VR-Link20017-1
  --rprFomVersion 2.0017 -F VR-Link20017-1.fed
```

- PC 5:

```
vrfSimHLA13 -s 1 -a 3004 -x VR-Link20017-1
  --rprFomVersion 2.0017 -F VR-Link20017-1.fed
```

5.2. The VR-Forces Window

VR-Forces shows all the simulated entities and interactions on a terrain database. The VR-Forces window has the following main components (Figure 5-5):

- Menus. Menus for accessing VR-Forces commands.
- Simulation control toolbars:
  - Simulation Control toolbar. Allows you to start, pause, and rewind a scenario.
  - Simulation Selection toolbar. Displays site and application numbers of multiple back-ends, if present.
  - Simulation Time toolbar. Displays the elapsed simulation time.
  - Simulation Time Scale toolbar. Allows you to control the speed at which the simulation runs.
- Application panels. Dockable panels that provide information and let you manage the observer.
- Convenience toolbars. Toolbars that let you quickly select frequently used menu commands and configuration settings.
- Object creation palettes. Palettes for creating entities, tactical graphics, and props.

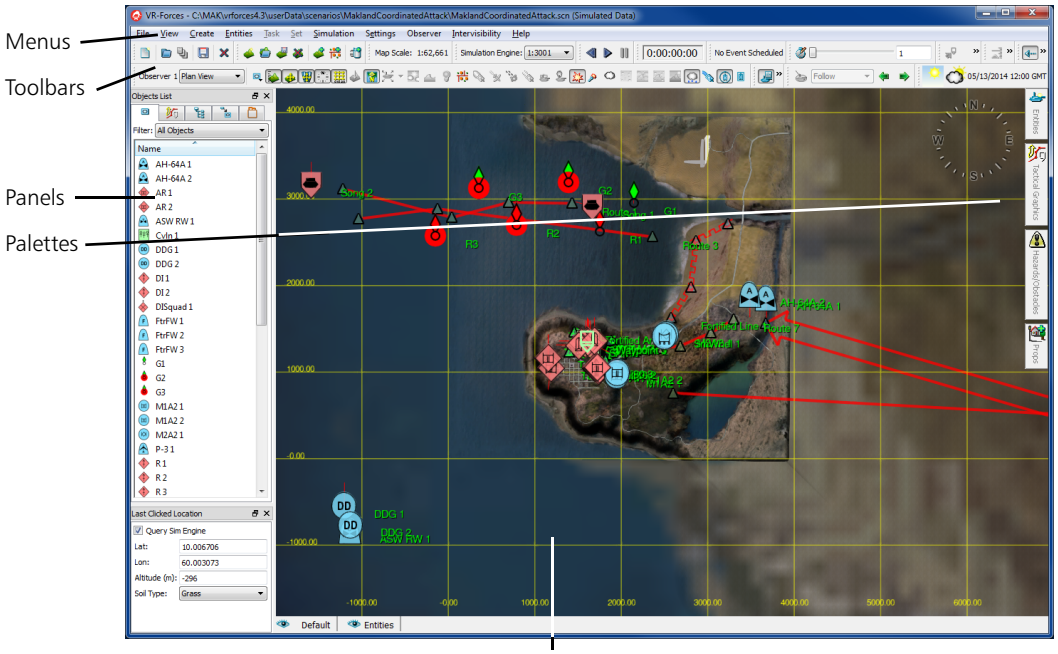


Figure 5-5. VR-Forces window and toolbars (2D)

You can view the terrain in 2D, 3D (Figure 5-6), and exaggerated reality (XR) (Figure 5-7) modes.

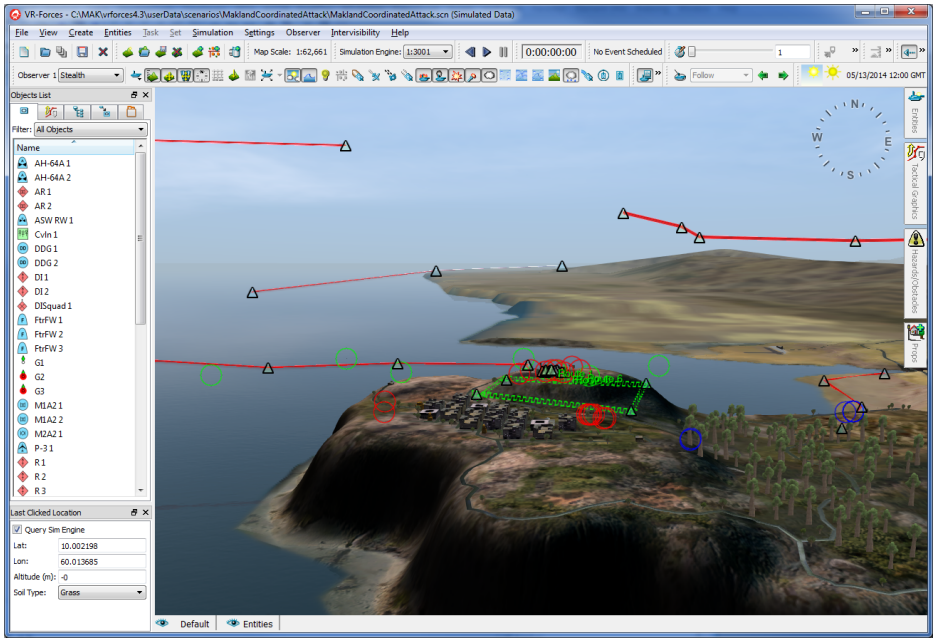


Figure 5-6. VR-Forces window (3D)

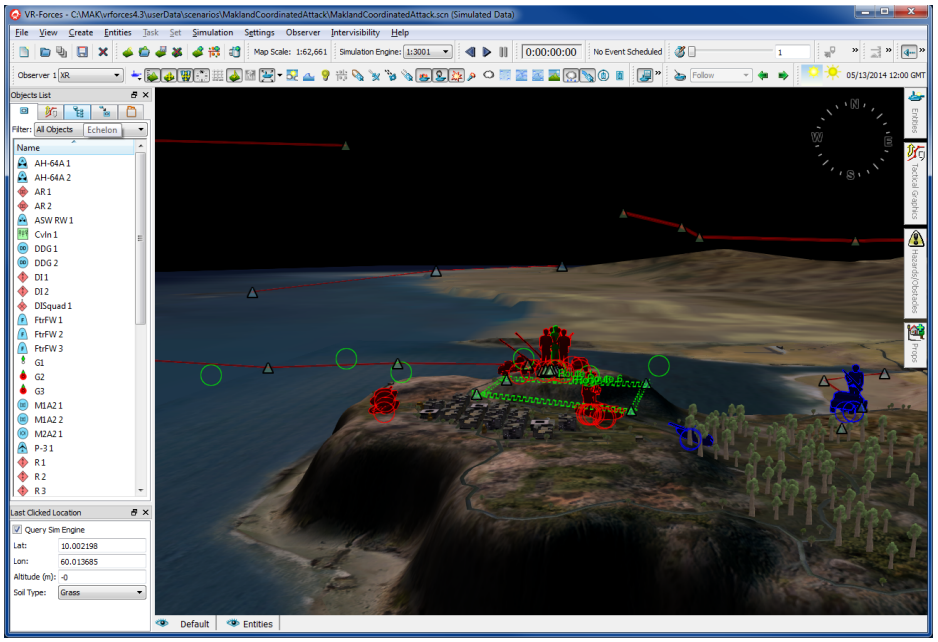


Figure 5-7. VR-Forces window (XR)

You can undock toolbars and panels and display them anyplace on your desktop. You can hide any toolbar that you do not want to be visible. (For details, please see Appendix A, *Common GUI Controls and Behaviors*.)

5.2.1. Opening New Windows

In addition to the main VR-Forces window, you can open secondary windows. These windows do not have any of the menus or toolbars of the main window. However, you can navigate the windows using the same keyboard and mouse options as the main window. When a secondary window has focus, the settings on the Observer Settings toolbar apply to that secondary window and changes you make affect the window.

You can open windows with a 2D view or 3D view. Once a window is open, you can change the view by changing the observer mode.

Opening a secondary window is not the same thing as running a second instance of the front-end. When you close a front-end, all secondary windows associated with it are also closed.

➤ To open a new window, choose **File** → **New Stealth Window** or **File** → **New Plan View Window**.



You can also open a new window by opening an inset view. For details, please see "Inset Views," on page 8-7.

5.2.2. Printing the VR-Forces Window

You can print the VR-Forces window to any printer configured on your computer. Only the display window gets printed. None of the GUI controls or tabs get printed.

To print the VR-Forces window:

1. Choose **File** → **Print**. The Print dialog box opens.
2. Select a printer.
3. Click Print.

5.3. Managing a Front-end's Session Connection

A front-end can be joined to a session or not joined. When a front-end is not joined to a session, it cannot create, load, or manage a scenario. It can only load a terrain and function as a viewer. You can tell a front-end's session status by the availability of Join Session and Resign Session commands on the File menu.

By default, when you start a front-end it automatically tries to join a session. If there is no session running, it will poll the network until one becomes available and then automatically join it. Once a front-end is joined to a session, you can manually resign from and rejoin it.

If you have one front-end running, when you open and close a scenario, the front-end opens and closes the terrain database for the scenario, as you would expect. If you have two or more front-ends loaded, the front-end from which you load a scenario automatically loads the terrain. However, all other front-ends prompt you to load the terrain. (The front-end knows which terrain to load, it is just asking permission to do so.) Similarly, when you close a scenario, the front-end from which you close the scenario automatically closes the terrain. All other front-ends ask permission to close the terrain. In either of these cases, you do not have to accept the default behavior. You can load an alternative terrain when a scenario opens and you can keep the terrain loaded when a scenario closes.

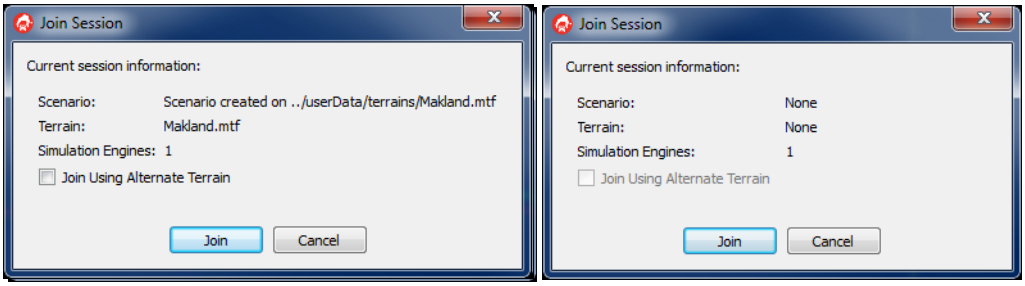
If you do not load the terrain specified by the scenario, it is possible that visual and simulation terrains will not be synchronized.

5.3.1. Joining a Session

When you join a session, all menu commands related to scenario management are enabled. If a scenario is loaded in the session, the terrain is automatically loaded in the joining front-end.

To join a session:

1. Choose **File** → **Join Session**. If VR-Forces is configured to always join a session with the session database, the front-end loads the session database and joins the session. If VR-Forces is not configured to automatically use the session database, the Join Session dialog box opens (Figure 5-8). If a scenario is running, its details are listed.



Scenario running | No scenario running

Figure 5-8. Join Session dialog box

2. If a scenario is running, you can specify that you want to load an alternate terrain:
   a. Select the Join Using Alternate Terrain check box.
   b. Click Join. An Open Database dialog box opens.
   c. Select the terrain that you want to load.
   d. Click Open. The terrain is loaded and the front-end joins the session.

! If you join a session using an alternate terrain, the front-end (visual) and back-end (simulation) terrains might not be synchronized.

---

3. Click Join. The front-end joins the session.

5.3.2. Resigning from a Session

When you resign from a session, all menu commands related to scenario management are disabled. If you have a scenario loaded, the terrain is closed.

➤ To resign from a session, choose **File** → **Resign From Session**.

5.3.3. Configuring Session Messages and Join at Startup

You can configure the display of session-related messages and whether or not the front-end tries to join a session or open a terrain database automatically at startup.

To configure sessions:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the Session Options page (Figure 5-3).
3. Select the options that you want to enable, as follows:
   - Automatically Join Session. If checked, when the front-end starts, automatically join a session if it is available. If cleared, the front-end does not join a session and does not prompt you to do so.
   - Automatically Open Session Database without Joining Session. This option lets VR-Forces automatically load the terrain database for the current session and put you in view only mode. You cannot create objects or control them. It interacts with Automatically Join Session as follows:

Table 5-1: Join session and open session database interaction

Automatically Join Session	Open Session Database without Joining	Result
☑	☐	At startup, VR-Forces joins the session, loads the terrain, and you have full control of objects and scenario actions.
☐	☑	At startup, VR-Forces loads the terrain database. You have no ability to create published objects or control the scenario. (You can create local tactical graphics.) If you clear this checkbox after the terrain loads, it remains loaded.
☑	☑	If you resign from a session, the terrain database stays open and you cannot create published objects or control the scenario. (You can create local tactical graphics.)
☐	☐	If, after starting VR-Forces, you select Automatically Open Session Database without Joining Session and a session is open, the terrain is loaded.

- Join Session Options. You can select one of the following options for joining a session:
  - Ask to Join Session. Provides session data (Figure 5-8) and prompts you to approve whenever you join a session.
  - Always Join Session with Session Database. Always join a session using the database in the loaded scenario. (Does not display a Join Session dialog box.)
  - Always Join Session with Open Database. Always joins a session with whichever terrain is currently open in the front-end. This might generate a terrain correlation warning.
- Show Session Terrain Change Prompts. Enable if you want to get prompted when the terrain in a session changes
- Show Database Correlation Warning Dialogs. If enabled, VR-Forces displays a warning if the terrain in the front-end might not correlate with the terrain loaded in the back-end.
- Show Scenario Startup Dialog. Enable if you want be prompted when a new session starts.

4. Click Close.

5.4. Configuring Time Management for HLA Exercises

To use time management services in an HLA exercise, you must properly configure an RTI and enable time management in the VR-Forces back-end.

Time management is enabled at the application level, not the scenario level. Scenarios created without use of time management work in time-managed mode. No time management-specific information is saved as part of a scenario.

If you enable time management in the back-end, then whenever you create or load a scenario that specifies a fixed-frame mode (that is, the **frame-mode** parameter in the scenario file is set to either fixed-frame best effort or fixed-frame-run-to-complete), the back-end will become a regulating and constrained federate. It will remain in this state until the scenario is closed.

A time-managed back-end advances simulation time at a rate that is a function of the frame-mode specified in the scenario it is executing. If the **frame-mode** is fixed-frame, then it tries to advance simulation time at a rate approximating real time. However, if the back-end has a heavy simulation load, or another time-constrained federate is advancing at a relatively slower rate, the back-end may not achieve this rate. If the **frame-mode** is fixed-frame-run-to-complete, the back-end tries to advance simulation time as fast as possible, independent of wall-clock time.

For conceptual information about time management, please see "Using HLA Time Management," on page 3-41.

5.4.1. RTI Requirements for Time Management

To use time management in VR-Forces, your RTI must be properly configured to support this service. If you are using the MÄK RTI, set the **RTI_useRtiExec** and **RTI_timeMgmt** parameters to 1. When you start the federation, you must run the rtiexec.

If you are using another RTI, please consult its documentation for configuration instructions.

5.4.2. Enabling Time Management for the Back-end

To enable time management in the back-end, use one of the following methods:

- Set **runInTimeManagementMode** to 1 in *vrfSim.mtl*.
- Start the back-end with the `--timeManagement` command-line option. For example, to start the back-end in time management mode using HLA 1.3, the sample FED file configured for use with time management, and the federation name VR-LinkTM:

```
vrfSimHLAprotocol -s 1 -a 3001 --timeManagement
  -F VR-LinkTM.fed -x VR-LinkTM
```



- The back-end will only run in time management mode (time regulating and constrained) while it has a scenario loaded.
- If a scenario is configured with **frame-mode** set to variable-frame, the back-end will not become time regulating or constrained. It will print a warning to the console and run without time management enabled.

5.5. Opening a Terrain Database

If you are using multiple front-ends, when you load or create a scenario, the front-end on which you issue the command loads the correct terrain database. The other front-ends prompt you to load the terrain. However, you can choose to load a different terrain. Also, if you want to use the front-end as a viewer, you can manually load a terrain database. (For a brief description of the databases supplied with VR-Forces, please see "Terrain Databases Provided with VR-Forces," on page 18-17.)

You can only "open" terrains in MTF format. To open any other terrain format, you must compose a terrain, as described in Chapter 13, *Composing Terrains*, in *VR-Forces Configuration Guide*.

The first time you load a terrain (including the terrains supplied with VR-Forces), VR-Forces caches textures in its fast load format. This is a slow process. However, thereafter the terrain will load quickly. If you want to avoid runtime caching, you can preprocess the terrain with the medfTool before you load it. If you want to disable runtime caching, you can do so as described in Section 13.9, "Configuring File Caching," in *VR-Forces Configuration Guide*.

It is possible to open a terrain without having a scenario loaded. In such cases you cannot create any type of VR-Forces object except tactical graphics. To create and manipulate entities and objects, you must load or create a scenario.

To open a terrain database:

1. In the VR-Forces window, choose **File** → **Open Terrain**. The Open Terrain dialog box opens.
2. Select the terrain that you want to open.
3. Click Open. The database is displayed in the VR-Forces window (Figure 5-9).

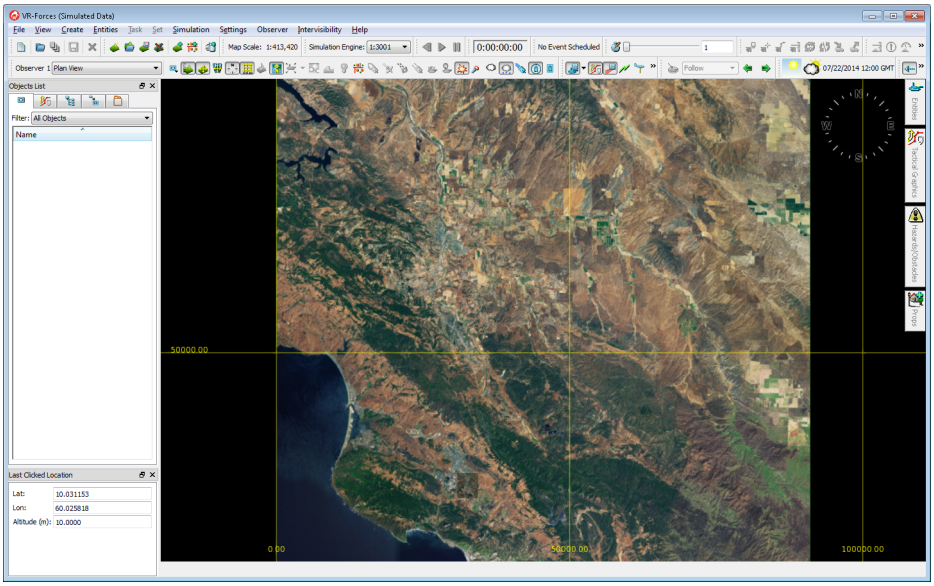


Figure 5-9. VR-Forces window with a database loaded

- If you load a terrain database while one is currently loaded, VR-Forces closes the current database and then opens the new one.
- Paged terrains might not be visible when loaded in Plan View mode. To see the terrain, press **e** to zoom in. This problem potentially exists for all paged terrains whose page-in distance is less than the default observer distance in Plan View mode.
- Some MetaFlight terrains, such as HarrisAfghanVillageFlatEarthMetaflight and HarrisAfghanVillageGeocentricMetaFlight do not display in the 2D view when first loaded. To see the terrain, change to a 3D view mode, and press the spacebar (center view on terrain) to move the eyepoint to the terrain location. After this, the terrain can be viewed in the 2D view.

5.5.1. Loading a Terrain Database at Startup

You can load a terrain database at startup by specifying it on the command line. If you start VR-Forces while a session is running, it can load the appropriate terrain database automatically. For details, please see "Configuring Session Messages and Join at Startup," on page 5-16.

➤ To load a terrain database when you start VR-Forces, use the `-T` command line option.

```
vrfSimprotocol -s 1 -a 3001 -T "../userData/terrains/myDa-
   tabase.mtf"
```

where *`protocol`* is `DIS`, `HLA13`, `HLA1516`, or `HLA1516e`.

5.5.2. Saving a Terrain Database in GDB Format

VR-Forces lets you save terrains in the GDB format.

To save a database in the GDB format from the VR-Forces GUI:

1. Choose **File** → **Export GDB**. The Export GDB dialog box opens.
2. Type a name for the database.
3. Click Save.

5.5.3. Closing a Terrain

To close a terrain, use one of the following methods:

- Choose **File** → **Close Terrain**.
- Choose **File** → **New Terrain**.
- Open a different terrain.

5.6. Managing VR-Forces Settings

VR-Forces has the following types of settings:

- Independent, or display, settings are the individual settings managed on the Settings menu and the pages of the Display Settings dialog box. Changing an independent setting, such as enabling track histories or wireframe mode, does not affect other settings.

  Independent settings are categorized as being global settings or observer-specific settings. Global settings apply regardless of the observer mode. Observer-specific settings can be different for each observer mode. For more information, please see "Global Settings and Observer-Specific Settings," on page 5-24.

- Dependent settings are the schemas, model definitions, element definitions, and entity type mappings that collectively determine how entities and objects are modeled in the scene. Changing a dependent setting, such as a model definition may affect settings that depend on it, such as an entity definition.

VR-Forces manages settings as follows:

- As you make changes to settings during a session, it saves the changes.
- It remembers the settings in effect when you exit and uses them as the default settings in your next session.
- During any particular session, you can revert to the default settings.
- You can save the current settings to a file.
- You can import saved settings.
- You can restore settings to those that were in effect when VR-Forces was first installed (factory settings).

Dependent settings have two additional options that independent settings do not have:

- You can save or restore a specific setting.
- You can merge settings from a file into the current settings.

The procedures for saving and restoring settings are the same regardless of the types of settings. You start the process by clicking a button on a settings dialog box page. However, the way that VR-Forces saves or restores settings depends on the type of setting, as follows:

- When you initiate a save or restore operation on independent settings, it only affects the settings on the dialog box page from which you started the save or restore. If you have changed a setting on another page, it is not affected.
- When you initiate a save or restore operation on dependent settings, all of the changes to all of the related settings get saved or restored. This ensures that the relationships do not get broken.

(Figure 5-10) illustrates the settings management buttons. The callout numbers are referenced in the list of button descriptions.



Figure 5-10. Settings buttons

- Revert to the default settings (1). Override any changes made to settings during this session.
- Restore the factory settings (2). Change settings to the settings used when VR-Forces was installed. (The factory settings and default settings may be different.)
- Import settings from a file (3). Replace the current settings with those stored in the selected file.
- Merge in settings from a file (4). (Only supported for dependent settings.) Merging works as follows:
  - Saved settings that do not have a match in the current settings are added.
  - Saved settings replace matching settings.
  - Current settings that are not replaced (that is, for which there is no match in the saved settings) remain valid.

  Figure 5-11 illustrates how merged settings work. Setting A in the saved settings has no match in current settings, so it is added to the merged settings. Current settings B_2 and C_2 are replaced with saved settings B_1 and C_1. Current setting D has no match in saved settings, so it is retained in the merged settings.

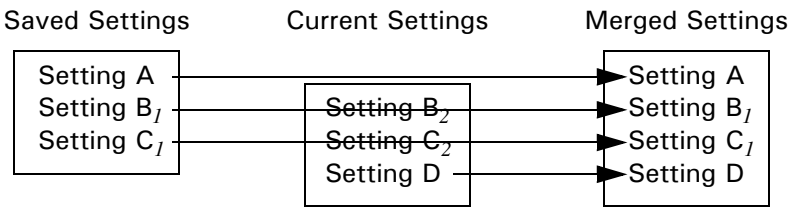


Figure 5-11. Merged settings

- Export settings to a file (5). Save the current settings to the selected file.
- Save the selected setting to a file (6). (Only supported for dependent settings.)

- The buttons described in this section are disabled on dialog box pages on which their function is not supported. This includes most of the pages on the Terrain Settings dialog box.
- VR-Forces saves settings to *./appData/settings/vrfGui*.
- If you load an alternative settings file at startup, it is treated like an imported settings file. If you save current settings as default, they get saved to the default files in *./appData/settings/*. They do not get saved back to the alternative file unless you explicitly export them to that file.

5.6.1. Synchronizing Settings Among VR-Forces Installations

If your simulation is using several front-ends and back-ends on multiple computers, you may want to use the same settings across all of them, particularly if you have added model definitions and mappings to the factory settings. To synchronize settings, copy the data directories (*./appData*, *./data*, and *./userData*) from the computer on which you have updated them to the computers on which the other front-ends and back-ends are installed. Then import the settings into any Settings page that you customized on the source VR-Forces machine.

5.6.2. Global Settings and Observer-Specific Settings

VR-Forces has two types of independent settings – global settings and observer-specific settings.

Global settings affect the display regardless of which observer mode you are using. Global settings include entity labels, tactical graphics, and cockpit displays. The settings on the Settings menu are all global settings.

Observer-specific settings can be different for each observer mode. In fact, given a similar projection and model set, it is the different settings that distinguish one observer mode from another. For example, one difference between Stealth observer mode and Out-the-Window observer mode is that Stealth mode has tactical smoke enabled and Out-the-Window mode has it disabled.

5.7. Exiting VR-Forces

When you exit VR-Forces from the GUI, all GUI windows close. In combined mode, the simulation engine also closes. If you are not running in combined mode, you must also close all simulation engines.

➤ To exit the VR-Forces GUI, choose **File** → **Exit**.

➤ To exit the simulation engine, type **`q`** and press **`Enter`**.

5.8. Configuring Simulation Connections

A simulation connection specifies the connection parameters for a DIS or HLA simulation connection. VR-Forces comes with the following connection configurations:

- DIS localhost (for use when disconnected from a network).
- DIS (port 3000)
- HLA 1.3 RPR 1.0
- HLA 1.3 RPR 2.0 DIGuy
- HLA 1516 RPR 1.0
- HLA 1516 RPR 2.0 DIGuy
- HLA 1516 Evolved RPR 2.0
- HLA 1516 Evolved RPR 2.0 Extended Aggregates.

You can edit the supplied configurations or add your own connection configurations that use, for example, a different DIS port or your own FOM Mapper.

5.8.1. Opening the Simulation Connections Configuration Dialog Box

Simulation connection configurations are maintained in the Simulation Connections Configuration dialog box. This dialog box opens automatically when you start VR-Forces, unless you choose a configuration as the default connection for auto connection. If Auto Connect is enabled and you want to change the connection, you can open the dialog box from the Tools menu.

➤ To open the Simulation Connections Configuration dialog box, on the Windows Start menu, choose **Programs** → **MAK Technologies** → **VR-Forces 4.3.1** → **Tools** → **Configure Connections**.

To start the vrfLauncher on Linux (or from the Windows command line), in a console window, enter the following command:

```
vrfLauncher options
```

For descriptions of the vrfLauncher command-line options, please see "Command-line Options for vrfLauncher," on page B-12.

5.8.2. Adding a Simulation Connection

To add a simulation connection:

1. If the Simulation Connections Configuration dialog box (Figure 5-12) does not open automatically when you start VR-Forces, choose **Programs** → **MAK Technologies** → **VR-Forces 4.3.1** → **Tools** → **Configure Connections**.

   To start the vrfLauncher on Linux (or from the Windows command line), in a console window, enter the following command:

   ```
   vrfLauncher options
   ```

   For descriptions of the vrfLauncher command-line options, please see "Command-line Options for vrfLauncher," on page B-12.

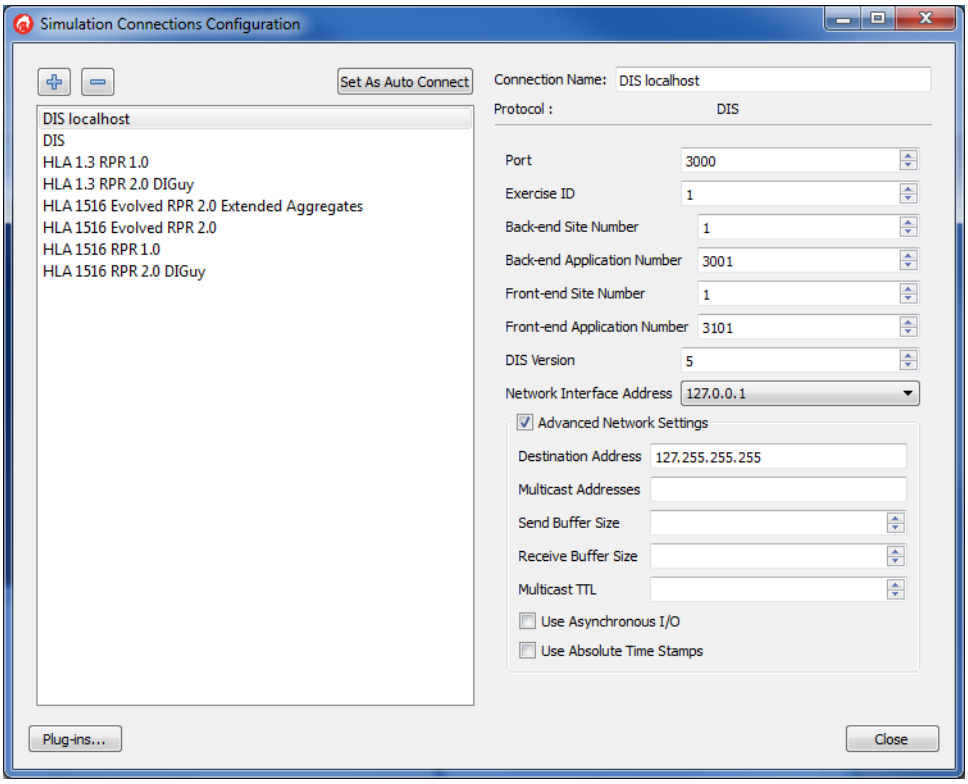


Figure 5-12. Simulation Connections Configuration dialog box

2. Click the Add Connection button (+). The Create Connection dialog box opens (Figure 5-13).

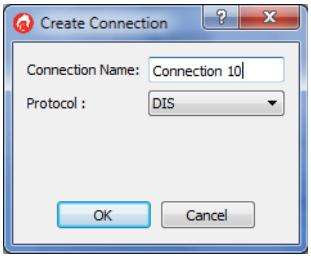


Figure 5-13. Create Connection dialog box

3. In the Connection Name box, type a name for the connection.
4. Select a protocol from the Protocol list.
5. Click OK.
6. The connection is added to the list in the Simulation Connections Configuration dialog box.
7. Select the new connection in the list.
8. Edit the parameters as desired. They are described in "Simulation Connection Parameters," on page 5-28.

5.8.3. Editing a Simulation Connection

To edit a simulation connection:

1. If the Simulation Connections Configuration dialog box (Figure 5-12) does not open automatically when you start VR-Forces, choose **Programs** → **MAK Technologies** → **VR-Forces 4.3.1** → **Tools** → **Configure Connections**.

   To start the vrfLauncher on Linux (or from the Windows command line), in a console window, enter the following command:

   ```
   vrfLauncher options
   ```

   For descriptions of the vrfLauncher command-line options, please see "Command-line Options for vrfLauncher," on page B-12.
2. In the list of connections, select the connection that you want to edit.
3. Edit the parameters as desired. They are described in "Simulation Connection Parameters," on page 5-28.

5.8.4. Simulation Connection Parameters

The HLA and DIS simulation connection parameters are equivalent to command-line options for DIS and HLA. Table 5-2 cross references the parameters to their descriptions in Appendix B, *Command-line Options*.

Table 5-2: Connection configuration parameters

Parameter	Command-line equivalent	Description
DIS		
Network Interface Address	`--deviceAddress`	Specifies the address of the network card to use for UDP traffic.
Port	`-P`	Specifies the DIS port. "Specifying the Port Number," on page B-18.
Exercise ID	`-x`	Specifies the DIS exercise ID. "Specifying the Exercise ID," on page B-19.
Back-end Application Number Front-end Application Number	`-a`	Specifies the application number. The default front-end application number is created by adding 100 to the back-end application number. "Specifying the Site ID and Application Number," on page B-13.
Back-end Site Number Front-end Site Number	`-s`	Specifies the site ID. These values can be the same or different. It is entirely arbitrary. "Specifying the Site ID and Application Number," on page B-13.
DIS Version	`--disVersion`	Specifies the DIS version: 4, 5, or 6. Default: 6. "Specifying the DIS Version," on page B-19.
Destination Address	`-A`	Specifies a specific destination address (point-to-point). "Specifying Point-to-Point or Multicast Operation," on page B-18.
Multicast Addresses	`-S`	Specifies one or more multicast addresses. "Specifying Point-to-Point or Multicast Operation," on page B-18.
Send Buffer Size	`--sendBufferSize`	Specifies the send buffer size. Default: -1.
Receive Buffer Size	`--recvBufferSize`	Specifies the receive buffer size.
Multicast TTL	`--mcastTtl`	Specifies the multicast time-to-live. Default: -1.
Use Asynchronous I/O	`--useAsyncIO`	Use asynchronous I/O. Default: False.
Use Absolute Time Stamps	`--useAbsoluteTimeStamps`	Specifies use of absolute time stamps instead of relative time stamps.

Table 5-2: Connection configuration parameters

Parameter	Command-line equivalent	Description
		! If you select Use Absolute Time Stamps, it is your responsibility to ensure that the time clocks of exercise participants are synchronized. If they are not synchronized, VR-Forces may not work properly.
HLA		
Network Interface Address	`--deviceAddress`	The IP address of the host computer.
Federation Name	`-x`	Specifies the name of the federation execution. "Specifying the Federation Execution," on page B-15.
FED File Name	`-F`	Specifies the FED file to use. "Specifying the FED File," on page B-16.
Back-end Application Number Front-end Application Number	`-a`	Specifies the application number. The default front-end application number is created by adding 100 to the back-end application number. "Specifying the Site ID and Application Number," on page B-13.
Back-end Site Number Front-end Site Number	`-s`	Specifies the site ID. These values can be the same or different. It is entirely arbitrary. "Specifying the Site ID and Application Number," on page B-13.
Ignore Advisories	`--useAdvisories`	Specifies whether or not to use HLA advisories.
Use Absolute Time Stamps	`--useAbsoluteTimeStamps`	Specifies use of absolute time stamps instead of relative time stamps.
		! If you select Use Absolute Time Stamps, it is your responsibility to ensure that the time clocks of exercise participants are synchronized. If they are not synchronized, VR-Forces may not work properly.
RPR FOM Version	`--rprFomVersion`	"Specifying the RPR FOM Version," on page B-16.
FOM Mapper Library	`-f`	Specifies the VR-Link FOM Mapper to use. "Specifying a FOM Mapper," on page B-16.
Initialization String	`--fomMapperInitData`	Specifies initialization data required by the FOM Mapper, if any. "Specifying FOM Mapper Initialization Data," on page B-17.

Table 5-2: Connection configuration parameters

Parameter	Command-line equivalent	Description
FOM Modules	`--fomModules`	Specifies FOM Modules. (HLA Evolved only.)
Local Settings Designator	`-s`	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.

5.8.5. Deleting a Simulation Connection

To delete a simulation connection:

1. If the Simulation Connections Configuration dialog box (Figure 5-12) does not open automatically when you start VR-Forces, choose **Programs** → **MAK Technologies** → **VR-Forces 4.3.1** → **Tools** → **Configure Connections**.
2. In the list of connections, select the connection that you want to delete.
3. Click the Delete button (−). A confirmation prompt opens.
4. Click Yes.

5.8.6. Configuring Auto Connect

If you know that you will usually or always want to use the same simulation connection, you can set up Auto Connect and you will not have to select a connection every time you start VR-Forces.

To configure Auto Connect:

1. If the Simulation Connections Configuration dialog box (Figure 5-12) does not open automatically when you start VR-Forces, choose **Programs** → **MAK Technologies** → **VR-Forces 4.3.1** → **Tools** → **Configure Connections**.
2. In the list of connections, select the connection that you want to use every time you start VR-Forces.
3. Click Set as Auto Connect. A star is displayed next to the selected connection.

Disabling Auto Connect

To disable Auto Connect:

1. Choose **Programs** → **MAK Technologies** → **VR-Forces 4.3.1** → **Tools** → **Configure Connections**. The Simulation Connections Configuration dialog box opens (Figure 5-12).
2. In the list of connections, select the connection that is configured for auto connect.
3. Click Set as Auto Connect. Auto connect is disabled and the star is removed from next to the connection.

5.8.7. Displaying Connection Information

You can display connection information while you are running VR-Forces. This may be useful for diagnostic purposes.

➤ To view connection information, choose **Settings** → **Connection Information**. The Connection Information dialog box opens (Figure 5-14).

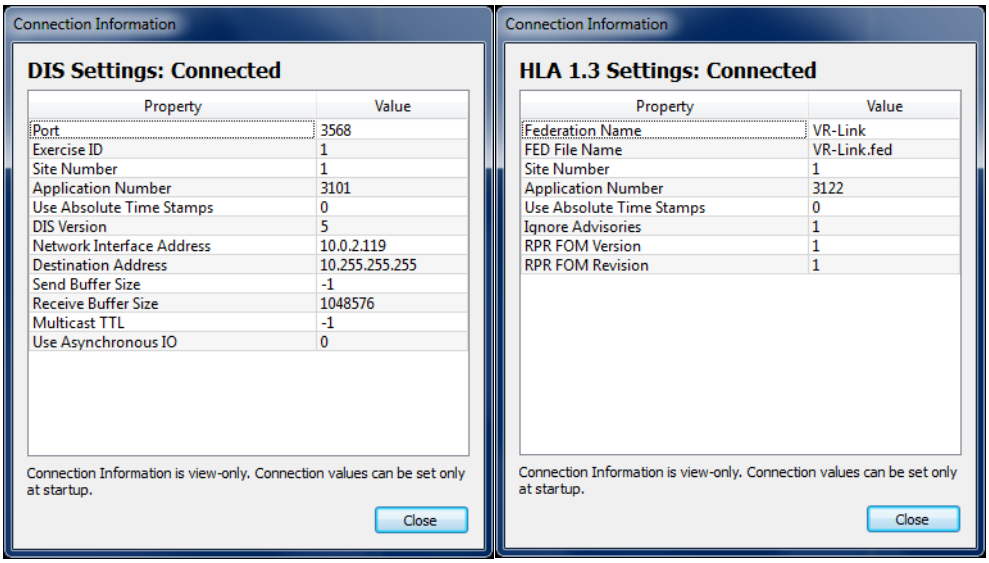


Figure 5-14. Connection Information dialog box

5.9. Managing Plug-ins

One way to extend the functionality of VR-Forces is by loading plug-ins that have been written for it by MÄK or other software vendors. The B-HAVE Module for VR-Forces is an example of a plug-in. You can manage which plug-ins get loaded when VR-Forces starts up. You can also view information about the plug-ins that have been loaded.

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-Forces application.

5.9.1. Loading Plug-ins

You can specify which plug-ins to load in either of the following ways:

- In the Plug-ins dialog box, specify the plug-ins to load the next time VR-Forces starts.
- In the Simulation Connections Configuration dialog box, specify the plug-ins to load at startup.

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-Forces.

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.
2. Select the Plug-ins Editor page (Figure 5-15).

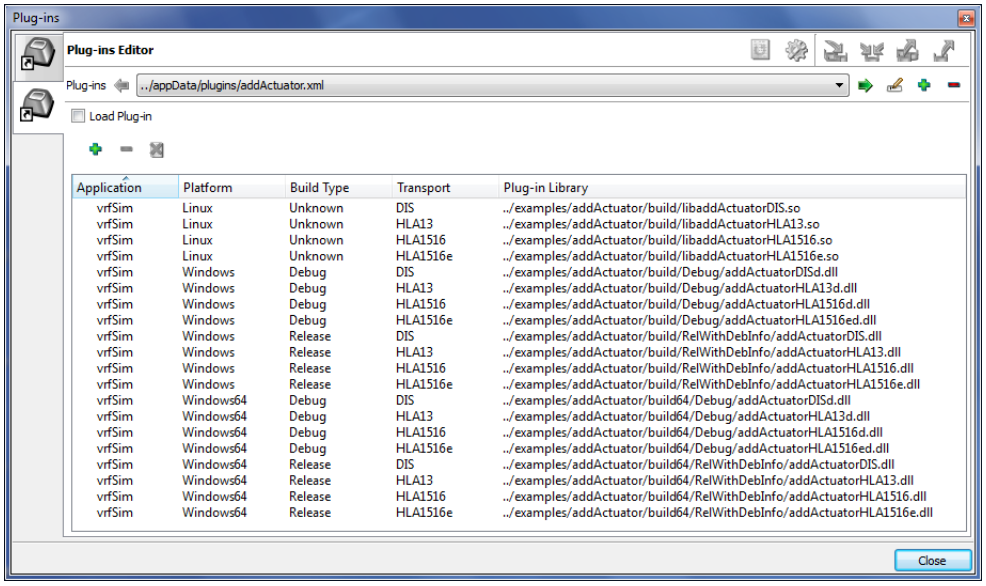


Figure 5-15. Plug-ins dialog box, Plug-ins Editor page

3. 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.
4. Select or clear the Load Plug-in check box.

Specifying the Plug-ins to Load in the Simulation Connections Configuration

You can select the plug-ins that you want to load at startup.

To specify the plug-ins to load in the simulation connections configuration:

1. Start VR-Forces using the VR-Forces Launcher, as described in "Starting VR-Forces from the VR-Forces Launcher," on page 5-4. The Simulation Connections Configuration dialog box opens (Figure 5-1).
2. Click Plug-ins. The Plug-ins Selection dialog box opens (Figure 5-16). The dialog box lists all of the plug-ins that are configured for VR-Forces. If a plug-in has previously been selected for loading, in this dialog box or in the Plug-ins Editor, it is checked.

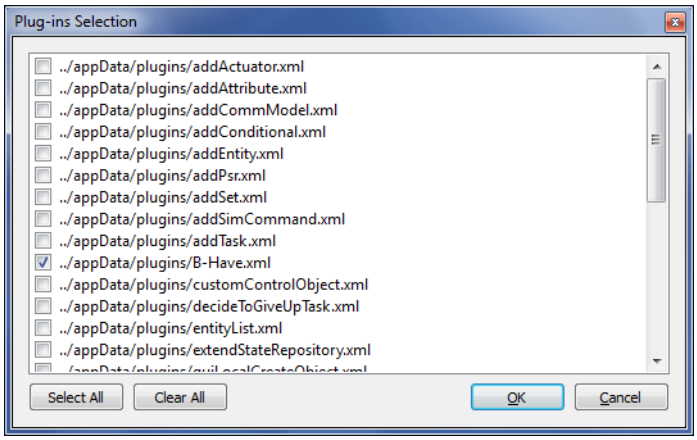


Figure 5-16. Plug-ins Selection dialog box

3. Select the check boxes for the plug-ins that you want to load. Clear the check boxes for any plug-ins that you do not want to load.
4. Click OK. The plug-in configuration applies to all of the startup configurations.

5.9.2. Adding a Plug-in

When you add a plug-in, VR-Forces creates an XML file (in the *./appData/plugins* directory) that contains all of the information VR-Forces 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 5-15).
2. Select the Plug-ins Editor page.
3. Click the Add button ( ) at the upper right corner of the page (#1 in Figure 5-17). 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 5-17).

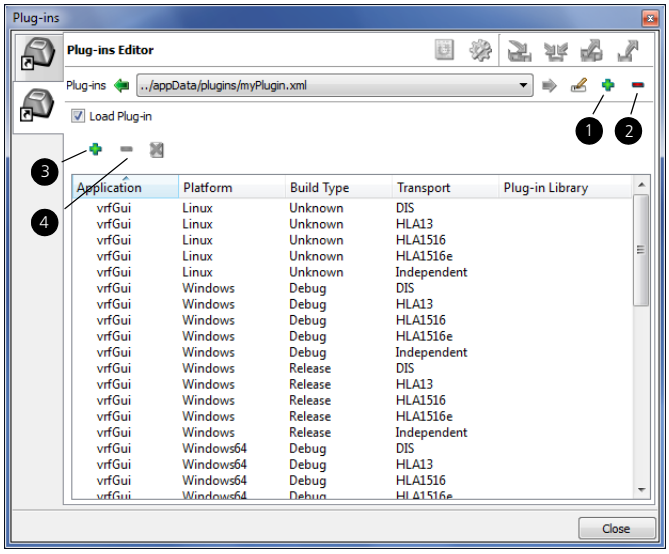


Figure 5-17. New plug-in

6. Specify the DLL for each configuration that you want to support, as described in "Specifying the DLLs for a Plug-in," on page 5-36.

5.9.3. 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 5-15).
2. Select the Plug-ins Editor page.
3. In the list of plug-ins at the top of the page, select the one that you want to edit.
4. Click the Plug-in Library column for the configuration that you want to edit. The Select Plug-in dialog box opens.
5. Select the DLL that you want to load for this configuration.
6. 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 5-15).
2. Select the Plug-ins Editor page.
3. Select the configuration whose DLL assignment you want to remove.
4. Click the Erase button ( ).

5.9.4. Adding a Plug-in Configuration

To add a plug-in configuration:

1. Choose **Settings** → **Plug-ins**. The Plug-ins dialog box opens (Figure 5-15).
2. Select the Plug-ins Editor page.
3. In the list of plug-ins at the top of the page, select the one that you want to edit.
4. Click the Add button (+) above the list of configurations (#3 in Figure 5-17). The Add Plug-in dialog box opens (Figure 5-18).

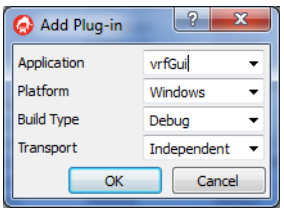


Figure 5-18. Add Plug-in dialog box

5. Select a values from the lists for Application, Platform, Configuration, and Transport.
6. Click OK. The configuration is added to the list.
7. Specify a DLL for the configuration, as described in "Specifying the DLLs for a Plug-in," on page 5-36.

5.9.5. Deleting a Plug-in Configuration

To delete a plug-in configuration:

1. Choose **Settings** → **Plug-ins**. The Plug-ins dialog box opens (Figure 5-15).
2. Select the Plug-ins Editor page.
3. In the list of plug-ins at the top of the page, select the one that you want to edit.
4. Select the configuration that you want to delete.
5. Click the Delete button (−) that is above the list of configurations (#4 in Figure 5-17).

5.9.6. 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 5-15).
2. Select the Plug-ins Editor page.
3. In the list of plug-ins at the top of the page, select the one that you want to delete.
4. Click the Delete button (−) to the right of the plug-ins list (#2 in Figure 5-17).

5.9.7. Viewing a List of Loaded Plug-ins

You can view a list of the plug-ins that are loaded when you start up VR-Forces. Plug-ins may include a description that you can also view.

To view the details of loaded plug-ins:

1. Choose **Settings** → **Plug-ins**. The Plug-ins dialog box opens (Figure 5-19).
2. Select the Plug-ins Loaded page (Figure 5-19).

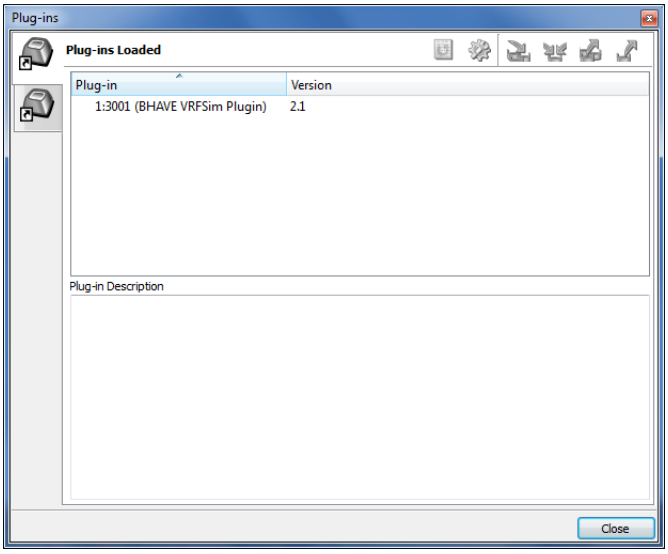


Figure 5-19. Plug-ins dialog box, Plug-ins Loaded page

3. To view details about a plug-in, select it in the list.

5.10. The VR-Forces Log Files

On Windows, VR-Forces creates two log files, *vrfSim.log* and *vrfGui.log*. Both log files are written to the directory from which you run VR-Forces.

On Linux, in combined mode, VR-Forces automatically creates a log file, *vrfSim.log* for back-end output. Front-end output is sent to the console. If you want to view back-end output, open a second console window, and do `tail -f vrfSim.log`. If you want to send front-end output to log files, or if you are running in separate mode and want to create a log file for either the front-end or back-end, you need to redirect your output to a file, for example:

```
vrfGui > vrfGui.log
```



6. Managing Objects

This chapter describes procedures and features that are common to entities and tactical graphics.

The Objects List Panel 6-2
Selecting Entities, Tactical Graphics, and Props 6-4
Selecting Objects in the Window 6-5
Selecting Objects in the Objects List Panel 6-6
Selecting an Entity in the Object Console Summary Panel 6-6
Unselecting Objects 6-7
Using Selection Groups 6-7
Creating a Selection Group 6-8
Selecting a Selection Group 6-9
Adding Objects to a Selection Group 6-9
Removing Objects from Selection Groups 6-10
Renaming a Selection Group 6-10
Importing and Exporting Selection Groups 6-10
Deleting a Selection Group 6-11
Displaying Tactical Graphics 6-11
Showing and Hiding Tactical Graphics 6-13
Showing and Hiding Tactical Graphics Labels 6-14
Displaying Height-Above-Terrain (HAT) Lines for Vertices (3D Only) 6-14
Displaying Information About an Object 6-15
Viewing Information for Multiple Objects 6-22
Viewing Object Counts 6-23

6.1. The Objects List Panel

There is a great deal of similarity in how you select, move, and view the different types of objects. You can manage them on the terrain, to some extent from the keyboard, and in the Objects List Panel, which is the central repository of object information.

The Objects List Panel has the following views:

- Entity list: displays entities in the simulation (Figure 6-1).
- Environmentals list: lists tactical graphics (control objects) (Figure 6-1).
- Echelon view: displays the entity hierarchy (Figure 6-2).
- Embarkation view: displays the embarkation relationships of entities (Figure 6-2).
- Overlays view: lists overlays and the objects that are on them (Figure 6-3).

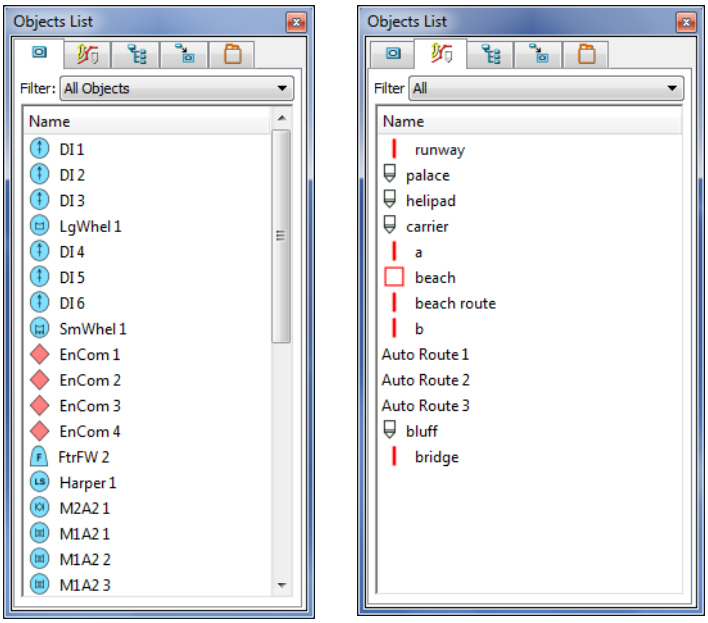


Figure 6-1. Objects List Panel – Entities view and Environmentals view

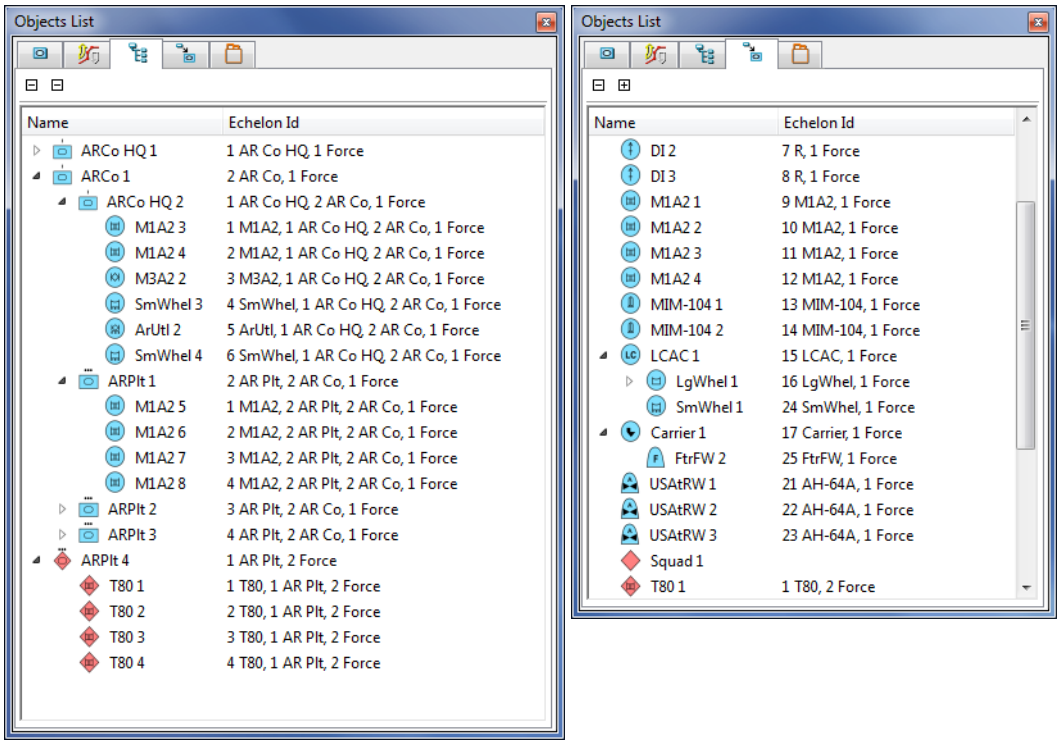


Figure 6-2. Objects List Panel – Echelon view and Embarkation view

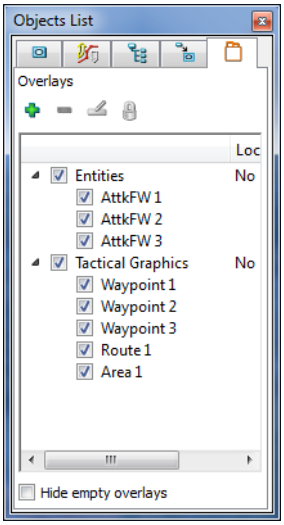


Figure 6-3. Objects List Panel – Overlays view

6.2. Selecting Entities, Tactical Graphics, and Props

VR-Forces displays the following kinds of objects that you might want to select:

- Entities. Any of the entity types that you can create.
- Cultural features. Objects in the Building and Cultural Feature categories on the Entities Palette.
- Tactical graphics. Control objects and the points, lines, areas, text, and symbols on the Tactical Graphics Palette, Hazards/Obstacles Palette, and Create menu.
- Props. Any of the props that you can create from the Props Palette or which are already part of the terrain.

You can select all types of objects in the window. You can select entities and tactical graphics in the Objects List Panel (Figure 6-1), you can select entities in the Object Console Summary Panel, and you can select props on the Props page of the Terrain Settings dialog box (Figure 6-4).

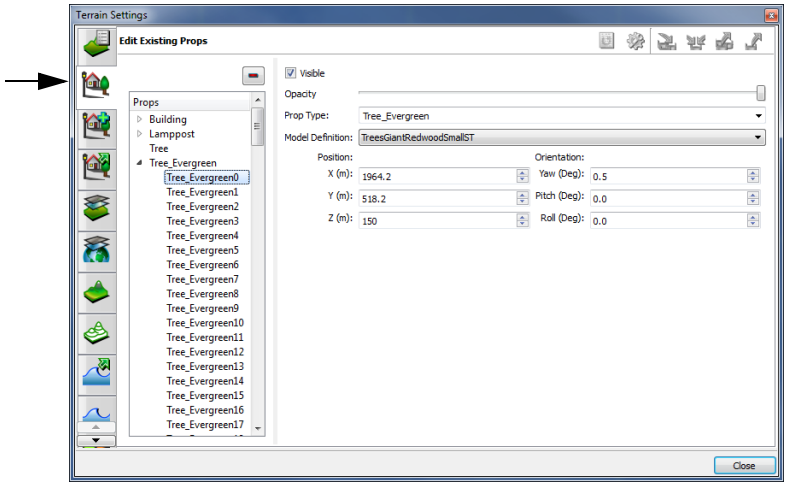
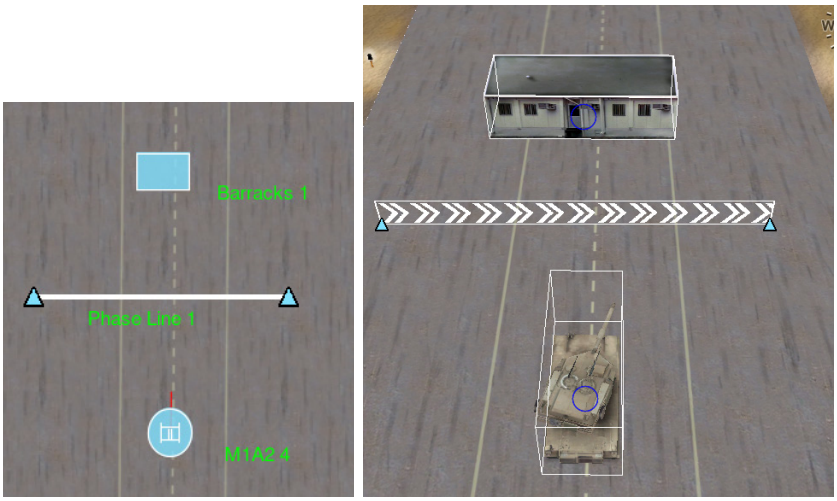


Figure 6-4. Terrain Settings dialog box, Props page

When you select an object in the 2D view, its border color changes to white. In the 3D views, all selected 3D objects display a bounding box (Figure 6-5). Tactical graphics change to white lines.



2D 3D

Figure 6-5. Selected objects

6.2.1. Selecting Objects in the Window

To select objects in the window, use any of the following mouse actions:

Mouse Action	Result
Click the object.	Selects an object (and unselects any other selected objects).
Ctrl-click the object.	Selects an object while keeping existing selections.
Double-click the object.	Selects the object and makes it editable. This is the equivalent of selecting an object and choosing **Entities** → **Edit**.
Alt-drag a bounding box.	Selects multiple objects by dragging a selection box around them.
Right-click the object.	If no objects are selected, right-clicking selects an object and displays a menu. If one or more objects are selected, right-clicking a selected object maintains the selection and displays a menu. Right-clicking an unselected object changes the selection to the object you right-clicked.

When you select an object in the window, the views on the Objects List Panel add a gray highlight to the entry for the object to indicate that it is selected.

6.2.2. Selecting Objects in the Objects List Panel

To select an object in the Objects List Panel:

1. Select a view in the Objects List Panel (Figure 6-1).
2. Locate the object you want to select. You can filter the list to make it easier to find particular objects. (For details, please see "Filtering the Object Lists," on page 6-6.) Disaggregate and expand aggregates if necessary. (For details, please see "Expanding and Collapsing Aggregates," on page 9-2 in this manual, and Section 4.6, "Triggering Aggregate State Transitions," in *VR-Forces Scenario Management Guide*.)
3. Click the entry for the object. The selected object is highlighted in the Objects List Panel and on the terrain.
4. Optionally, select multiple objects, as follows:
   - **`Shift`**-click to select a contiguous range of entities.
   - **`Ctrl`**-click to select multiple non-contiguous entities.

Filtering the Object Lists

You can filter the display in the Entities List and the Environmental List.

➤ To filter the display of objects, select a category to view from the list at the top of the tab (Figure 6-1).

6.2.3. Selecting an Entity in the Object Console Summary Panel

To select an entity in the Object Console Summary Panel:

1. Right-click a message from the entity you want to select.
2. On the popup menu, choose Select Entity.

6.2.4. Unselecting Objects

To unselect objects in the window, choose one of the options in Table 6-1.

Table 6-1: Unselecting objects on the terrain

Action	Result
Click in the window away from the selected object or objects.	Unselects all objects.
Right-click.	Exits edit mode, but object remains selected.
Click a selected object.	If multiple objects were selected, unselects all except the object you clicked.
Ctrl-click a selected object.	Unselects the object. If multiple objects were selected, does not unselect other objects.

To unselect objects on the Objects List Panel, choose one of the options in Table 6-2.

Table 6-2: Unselecting objects in the Objects List Panel

Action	Result
If one object is selected.	**Ctrl**-click the object.
If a group of objects is selected:	
Ctrl-click an object.	Unselects the object without affecting other selections.
Click an object.	Selects the clicked object and unselects all other objects.

6.3. Using Selection Groups

Selection groups are named subsets of the objects in an exercise. Their purpose is to allow you to quickly select specific objects without needing to locate them on the map or in the Objects List Panel and select them individually. Using selection groups can be helpful if you want to quickly select a group of objects that do not fit neatly into the default filter options on the Objects List Panel tabs, for example a mix of friendly and opposing forces and graphical objects. The group definitions are saved and loaded as part of the scenario.

An object can be in multiple selection groups.

6.3.1. Creating a Selection Group

You can create selection groups in any of the following ways:

- In the Selection Groups panel.
- From Selection Groups menu options.
- Using keyboard shortcuts.

To create a selection group in the Selection Groups Panel:

1. Choose **View** → **Selection Groups Panel**. The Selection Groups Panel opens (Figure 6-6).

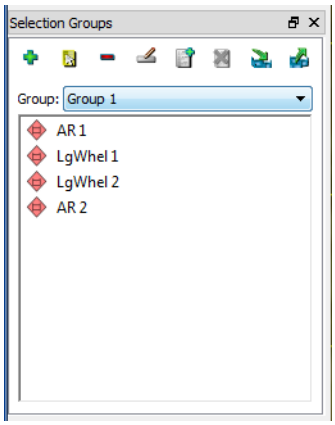


Figure 6-6. Selection Groups Panel

2. Click the Add button ( ). The New Selection Group dialog box opens.
3. Type a name for the selection group.
4. Click OK. The group is added to the group list.

Creating a Selection Group from a Selection Group Menu Command

When you create a selection group from a Selection Group menu command, you can combine the procedures for creating a new selection group and adding an object to it.

To create selection group from a Selection Group menu command:

1. Select the objects you want to put in a new selection group.
2. Choose **Entities** → **Selection Groups** → **Add to New Selection Group**. The New Selection Group dialog box opens.
3. Type a name for the selection group.
4. Click OK. The selection group gets created and the selected objects get added to it.

Creating a Selection Group from a Keyboard Shortcut

You can quickly create a selection group with a default name using a keyboard shortcut. When you create a selection group from the keyboard it is created as follows:

- If no objects are selected, an empty selection group is created.
- If objects are selected, a selection group is created with these objects as members.
- If the keyboard shortcut is for a selection group that already exists, a new selection group replaces the previous one. If the previous group had members and the new one is created without any objects selected, the new group is empty.

➤ To create a selection group from the keyboard, press **`Ctrl`** + *number*, where *number* is 1, 2, 3, 4, 5, 6, 7, 8, 9, or 0. A selection group is created named Selection Group *number* (0 creates Selection Group 10). (The keypad numbers do not support this operation.)

6.3.2. Selecting a Selection Group

When you select the objects in a selection group, they are selected in the Objects List Panel and in the display window. They are not selected in the Selection Groups Panel list.

To select a selection group, use one of the following methods:

- In the Selection Groups Panel, select a selection group in the Group list. Then, click the Select All Objects In Selection Group button ().
- Choose **Entities** → **Selection Groups** → **Select Selection Group** → *group_name*, where *group_name* is one of the groups listed on the pull right menu.
- If selection groups named with the format Selection Group *number* exist, pressing a number key selects the corresponding selection group. (The keypad numbers do not support this operation.)

6.3.3. Adding Objects to a Selection Group

To add objects to a selection group:

1. Select the objects you want to add to a selection group.
2. Do one of the following:
   - Choose **Entities** → **Selection Groups** → **Add To Selection Group** → *group_name*, where *group_name* is one of the groups listed on the pull right menu.
   - On the Selection Groups Panel, click the Add Current to Selection Group button ().

6.3.4. Removing Objects from Selection Groups

There are two ways to remove an object from a selection group.

To remove an object from a selection group:

1. Select the objects you want to remove from a selection group.
2. Choose **Entities** → **Selection Groups** → **Remove from Selection Group** → *group_name*, where *group_name* is one of the groups listed on the pull right menu.

To remove an object from a selection group in the Selection Groups Panel:

1. Select the selection group from which you want to remove an object.
2. Select the object you want to remove.
3. Click the Remove Selected from Selection Group button ( ).

6.3.5. Renaming a Selection Group

To rename a selection group:

1. Choose **View** → **Selection Groups Panel**. The Selection Groups Panel opens (Figure 6-6).
2. In the Group list, select the group you want to rename.
3. Click the Rename button ( ). The New Selection Group Name dialog box opens.
4. Type a new name for the selection group.
5. Click OK.

6.3.6. Importing and Exporting Selection Groups

When you save a scenario, selection groups gets saved to the *scenario_name.sgr* file. You can also export selection groups separately and import them into other scenarios. When you import a selection group file, it replaces any selection groups that are currently in the scenario.

To export a selection group:

1. Choose **View** → **Selection Groups Panel**. The Selection Groups Panel opens (Figure 6-6).
2. In the Group list, select the selection group you want to export.
3. Click the Export button ( ). The Export Selection Groups dialog box opens.
4. Type a name for the selection group file.
5. Click Save.

To import a selection group:

1. Choose **View** → **Selection Groups Panel**. The Selection Groups Panel opens (Figure 6-6).
2. Click the Import button (). The Import Selection Groups dialog box opens.
3. Select the selection group file that you want to import.
4. Click Open.

6.3.7. Deleting a Selection Group

Deleting a selection group does not delete the objects from the scenario.

To delete a selection group:

1. In the Selection Groups Panel, select the group you want to remove.
2. Click the Delete button ().

6.4. Displaying Tactical Graphics

Tactical graphics are graphical objects that represent points, lines, areal objects, text, and symbols used for various purposes in simulations (Figure 6-7 and Figure 6-8). They could represent routes that entities are following, a destination point, or a way to show some sort of relationship, such as height-above-terrain (HAT) lines.

You can configure the display of tactical graphics as follows:

- Enable and disable the display of tactical graphics.
- Show or hide vertex indicators for point, linear, and areal objects.
- Show or hide height-above-terrain lines.
- Display height-above-terrain lines only for selected graphics.

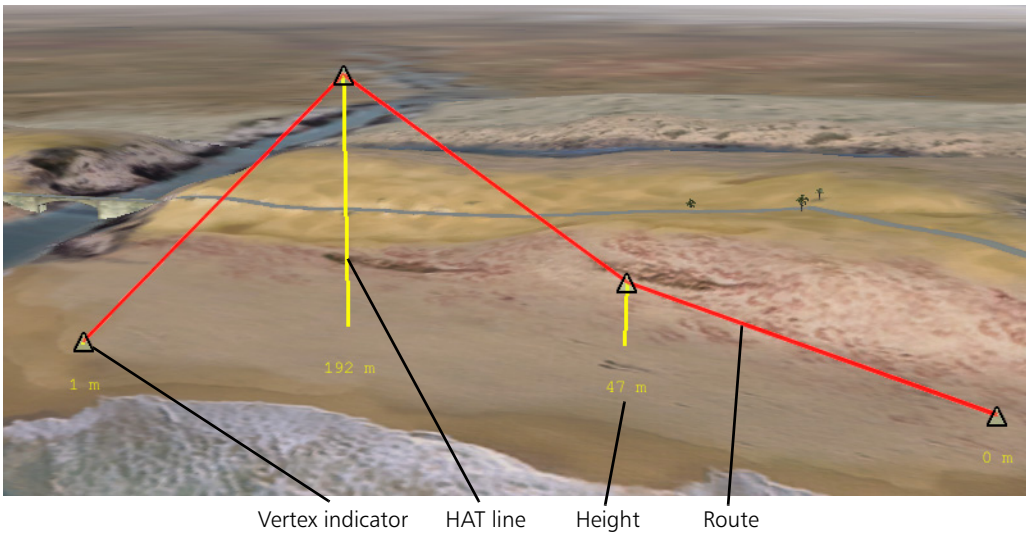


Figure 6-7. Tactical graphics with height-above-terrain lines (3D)

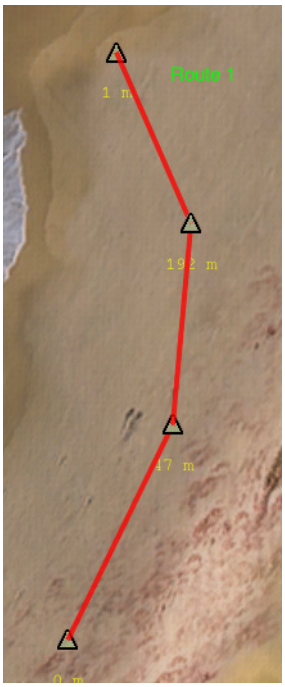


Figure 6-8. Tactical graphic (2D)

6.4.1. Showing and Hiding Tactical Graphics

You can control the display of tactical graphics from the Display Settings dialog box, Tactical Graphics Display Settings page, from menu and toolbar options, and from the Overlays tab.

To show or hide tactical graphics:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Tactical Graphics Display Settings page (Figure 6-9).

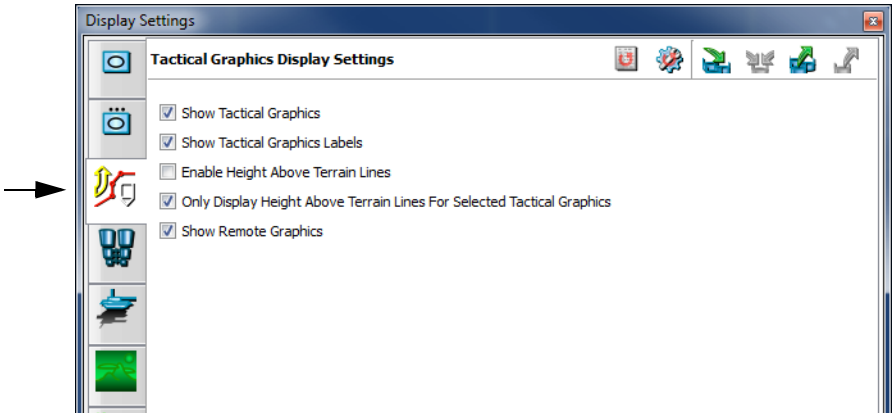


Figure 6-9. Tactical Graphics Display Settings page

3. Select or clear the Show Tactical Graphics check box.

> You can also enable and disable tactical graphics by choosing **Settings** → **Tactical Graphics** or clicking the Tactical Graphics button ( ) on the Display Settings Toolbar.

Showing and Hiding Tactical Graphics on the Overlays Tab

On the Overlays tab, you can show and hide tactical graphics individually and by overlay.

To control the display of tactical graphics from the Overlays tab:

1. On the Objects List Panel, select the Overlays tab (Figure 6-3).
2. To hide all graphics on a particular overlay, clear the check box for that overlay.
3. To hide a specific tactical graphic, clear its check box.
4. To redisplay a hidden tactical graphic or overlay, select its check box.

6.4.2. Showing and Hiding Tactical Graphics Labels

In addition to displaying models for tactical graphics themselves, you can display graphics (labels) for the vertices of these objects. Vertex labels are enabled in Figure 6-7 and Figure 6-8. In Plan View mode, VR-Forces also displays a tactical graphic's name-label, and extended labels, if any. For details about displaying labels for 2D icons, please see "Displaying Entity Labels for 2D Icons," on page 7-7.



Displaying vertex labels can affect performance.

To show or hide vertex labels:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Tactical Graphics Display Settings page (Figure 6-9).
3. Select or clear the Show Tactical Graphics Labels check box.



You can also enable and disable tactical graphics labels by choosing **Settings** → **Tactical Graphics Labels** or clicking the Tactical Graphics Labels button () on the Display Settings Toolbar.

6.4.3. Displaying Height-Above-Terrain (HAT) Lines for Vertices (3D Only)

You can display height-above-terrain (HAT) lines for the vertices of tactical graphics. These vertical lines between an object's vertices and the ground help you understand the relationship of the object to the terrain. The height for each vertex is displayed (using whatever units are configured on the Display Units Settings page) (Figure 6-7).

HAT lines are shown only for tactical graphics that could be above or below the ground, such as routes, and waypoints. You can show and hide these lines and you can specify that they be shown for all objects or only for the selected objects.

Use of height above terrain lines can reduce the frame rate.

To show or hide height-above-terrain lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Tactical Graphics Display Settings page (Figure 6-9).
3. To display HAT lines, select the Enable Height Above Terrain Lines check box.
4. To only display lines for selected objects, select the Only Display Height Above Terrain Lines For Selected Tactical Graphics check box.

6.5. Displaying Information About an Object

You can display information about individual objects, including entities, aggregates, environmental objects, and Stealths. You cannot edit this information. You can also display entity labels. For details about entity labels, please see "Displaying Entity Labels," on page 7-3.

An information dialog box has the following sections:

- Summary.

  Displays the following information for entities in entity-level scenarios (Figure 6-12):

  – Entity icon.
  – Entity name.
  – Current task.
  – Sensor information.
  – Speed.
  – Altitude.
  – Health and weapon status.

  Displays the following information for entities in aggregate-level scenarios (Figure 6-13):

  – Entity icon.
  – Entity name.
  – Current task.
  – Sensor information.
  – Posture.
  – Health status.
  – Health and weapon status gumballs.
- Detailed information.
- Object console.

The summary section for aggregates in aggregate-level scenarios (Figure 6-15) includes gumball icons that display entity health information. Entity icons also display gumballs.

Figure 6-10 describes the entity status represented by different gumball states.

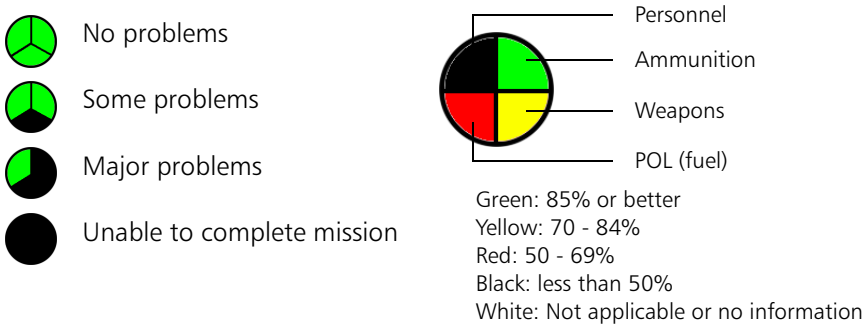


Figure 6-10. Gumball charts

Figure 6-11 illustrates gumball charts for an entity icon.



Figure 6-11. Entity icon with gumballs

You can expand and contract the detailed information and object console sections to manage the amount of screen space used by the dialog box.

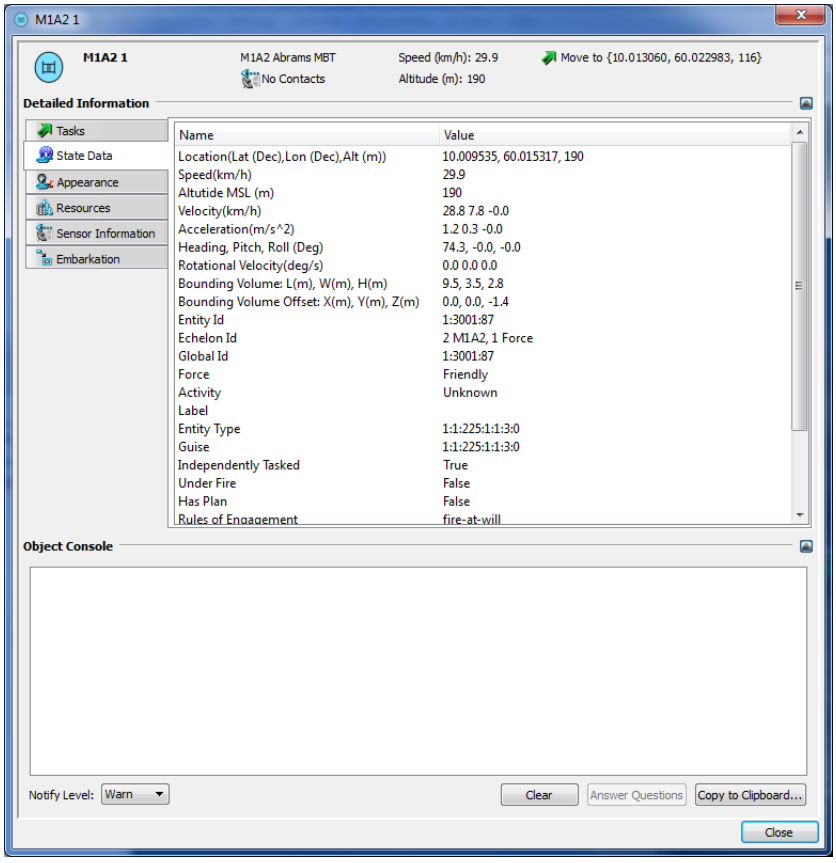


Figure 6-12. Information dialog box for entity in entity-level scenario

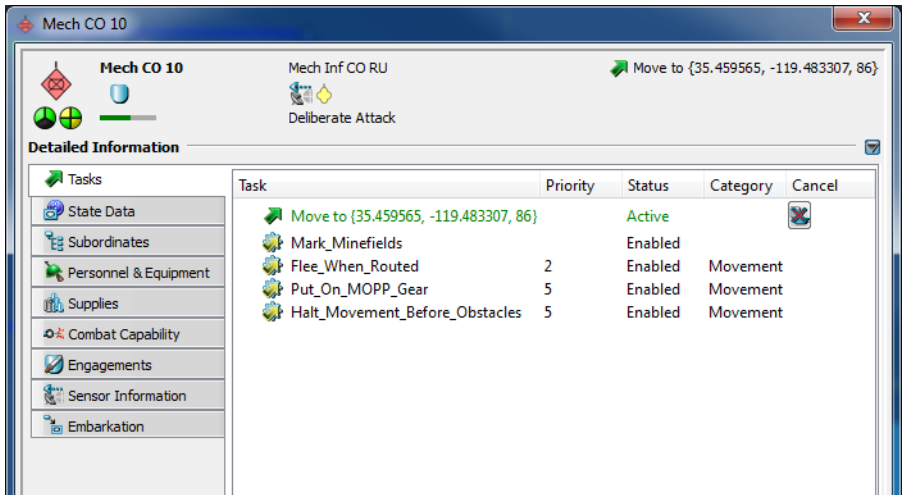


Figure 6-13. Information dialog box for an aggregate in aggregate-level scenario

The Detailed Information section for entities in an entity-level scenario has the following pages:

- Tasks. Shows the currently assigned task and any enabled reactive tasks. Allows you to cancel the current task.
- State Data. Shows location, movement values such as speed and heading, identifying information, and other status information.
- Appearance. Shows the status of appearance attributes, such as lights, firepower, and concealment.
- Resources. Shows that amount of fuel and ammunition available.
- Sensor Information. Shows information about entities being detected. For entities in aggregate-level scenarios, the page shows if the entity is being jammed (electronic warfare) or is jamming other entities.
- Emitters. Shows details about emitters, if applicable.
- Acoustics. Shows details about active sonar and propulsion noise, if applicable.
- Embarkation. Lists embarked objects, if any.
- IFF. Shows IFF information, if applicable.



The pages listed for Detailed Information are context sensitive. If a page does not apply to an entity type, it is not shown.

The Detailed Information section for aggregates in entity-level scenarios (Figure 6-14) has the following pages:

- Tasks.
- State Data.
- IFF.
- Subordinates. Lists the subordinates.
- Acoustics.
- Embarkation.

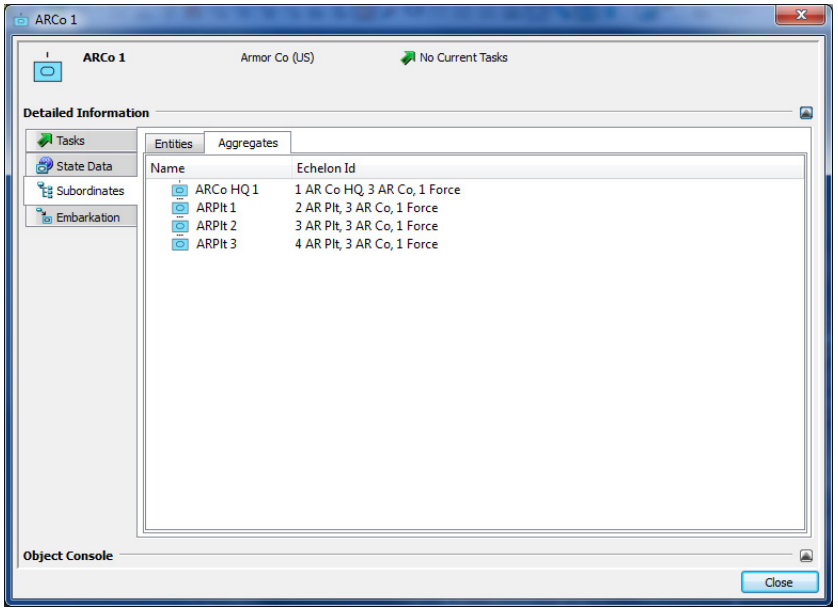


Figure 6-14. Information dialog box for aggregates in entity-level scenarios

The Detailed Information section for entities in aggregate-level scenarios has the following pages:

- Tasks.
- State Data.
- IFF.
- Subordinates. Lists the subordinates.
- Personnel and Equipment. The types of personnel and number of each. Casualties, by category.
- Supplies. Food, water, fuel, and ammunition. The Suppliers tab displays information when an entity is receiving supplies or providing them to another entity. When an entity is receiving supplies, an up arrow is displayed next to the supply it is receiving.
- Combat Capability. Combat power by category (anti-air, anti-personnel, and so on) and defensive capability, by category.
- Engagements. Current attack and defense status.
- Sensor Information. The Sensors tab shows information about entities being detected. The Jamming tab shows if the entity is being jammed (electronic warfare) or is jamming other entities.
- Emitters.
- Acoustics.
- Embarkation.

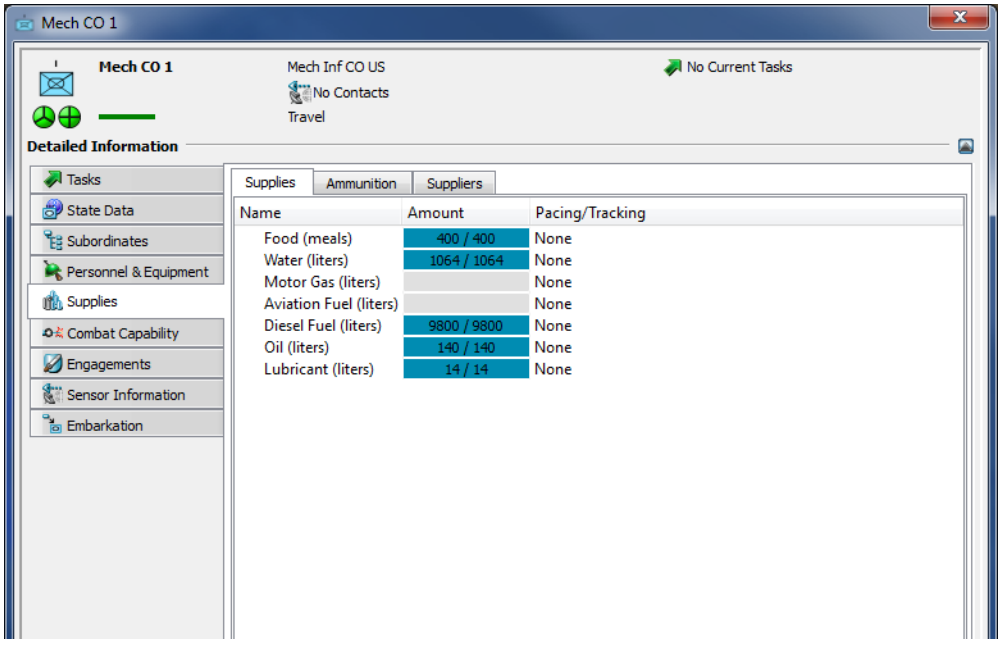


Figure 6-15. Information dialog box for entity in aggregate-level scenario

The detailed information section for an environmental object (Figure 6-16), such as a route, has the following pages:

- State Data. Shows the force and coordinates.
- Attached To. The entity this object is attached to, if any.

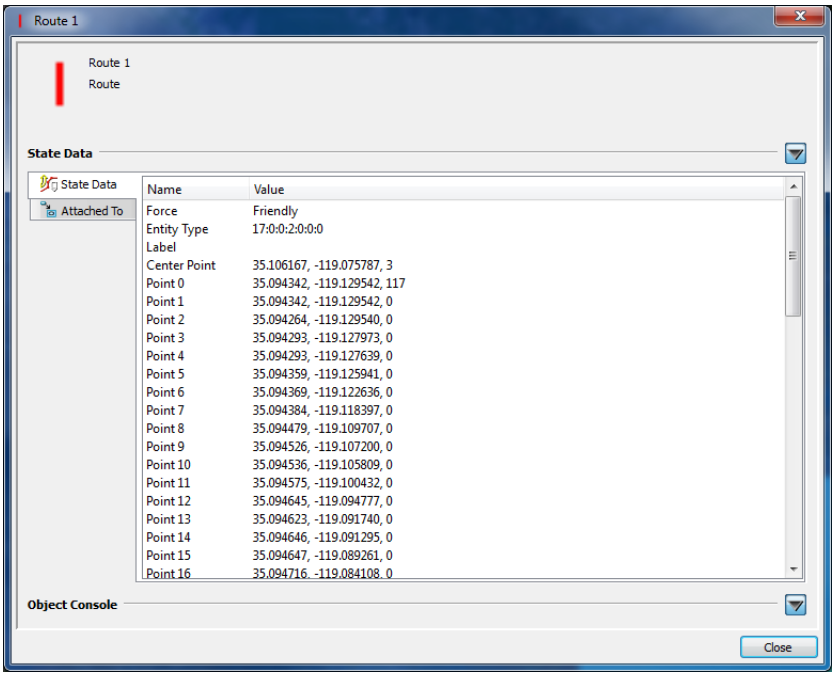


Figure 6-16. Environmental object information dialog box

To display an object information dialog box:

1. Select an object.
2. Choose **Entities** → **Information**, or press **`i`**.



You can also open the Entity Information dialog box by selecting a message in the Objects Console Summary Panel and choosing **Information** on the popup menu.

6.5.1. Viewing Information for Multiple Objects

You can select multiple objects and view summary information for all of them. From the summary information window, you can jump directly to an object and you can open its Information dialog box.

To view information about multiple objects:

1. Select the objects.
2. Choose **Entities** → **Information**, or press **i**. The Multiple Object Information window opens (Figure 6-17).

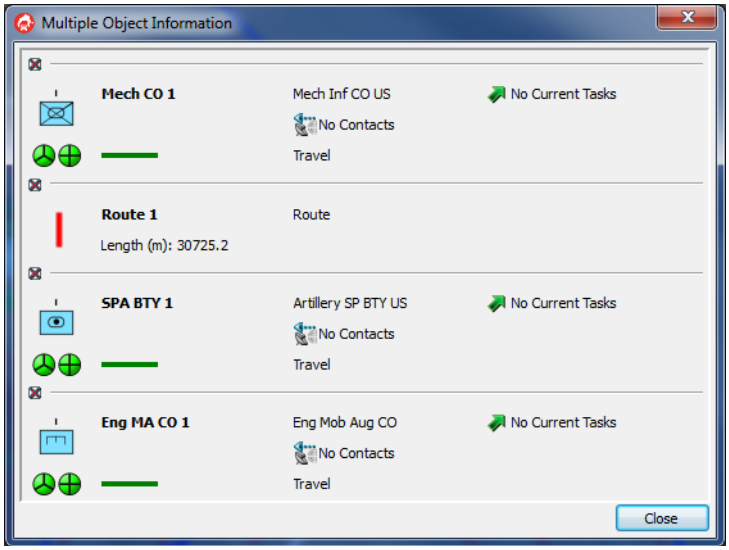


Figure 6-17. Multiple Object Information window

3. To zoom to the extents of an object, click its icon.
4. To open an Information dialog box for an object, click any of the details about the object, such as its name.

6.6. Viewing Object Counts

You can display the number of live and destroyed entities and aggregates in the simulation you are viewing and the number of control objects, by type.

➤ To display the object counts, choose **View** → **Objects Count Panel**. The Objects Count Panel is added to the display (Figure 6-18).

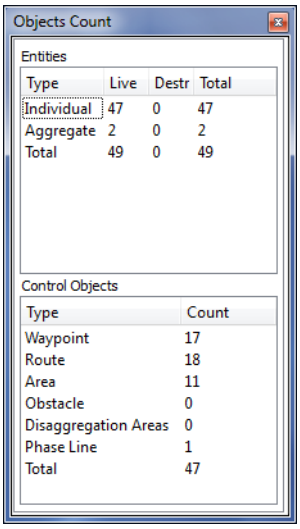


Figure 6-18. Objects Count Panel



7. Viewing Entities and Entity Information

This chapter describes how to get information about entities and how to attach the observer to entities.

Displaying Entity Labels 7-3
- Displaying Text Labels for 3D Models 7-5
- Turning Entity Labels On and Off 7-6
- Pinning 3D Text Labels to the Window 7-6
- Displaying Entity Labels for 2D Icons 7-7
- Configuring Shadowing and Background Color for 2D Labels 7-9

Using Extended Labels 7-10
- Adding an Extended Label 7-11
- Changing an Extended Label's Index 7-12
- Deleting an Extended Label 7-12

Entity Icons 7-13
- Changing the Size of 2D Icons 7-14
- Rotating 2D Icons to an Entity's Heading 7-15

Model Sets 7-15
- Changing the Model Set 7-15

Exaggerating the Scale of 3D Entity Models 7-16

Hiding 3D Models 7-19

Changing Force Colors 7-19

Displaying HAT Lines for Entities (3D Only) 7-20

Displaying Threat and Sensor Range Rings 7-21
- Enabling and Disabling the Display of Range Rings 7-23
- Displaying Range Rings for an Entity 7-23
- Configuring the Display of Range Rings 7-24

Displaying Entity Resources 7-26
Viewing Numerical Resource Data 7-26
Displaying Bounding Volumes 7-27
Showing Bounding Volumes Only for Selected Entities 7-28
Configuring Object Console Messages 7-28
Setting the Notification Level for Console Messages 7-29
Viewing All Console Messages 7-30
Answering Questions from Scripted Tasks 7-31
Saving Console Messages 7-31
Displaying an Object Console Warning Icon 7-32
Clearing the Object Console 7-32

7.1. Displaying Entity Labels

VR-Forces provides extensive information about entities in the Entity Information dialog box ("Displaying Information About an Object," on page 6-15). It also provides an abbreviated set of information in the form of entity labels.

For 3D models, VR-Forces displays entity labels that help you locate and identify entities (Figure 7-1). The entity label consists of an entity locator circle, which is always visible when entity labels are enabled, and a text label that is displayed when you mouse over an entity. All entities within the view have locator circles, whether the model is visible or not. The color of an entity's label represents its force – blue for friendly, red for opposing, and green for neutral.

A 3D text label displays the entity's:

- Marking text
- Entity type enumeration
- Force
- Name
- Speed
- Heading
- Latitude and longitude
- Altitude.

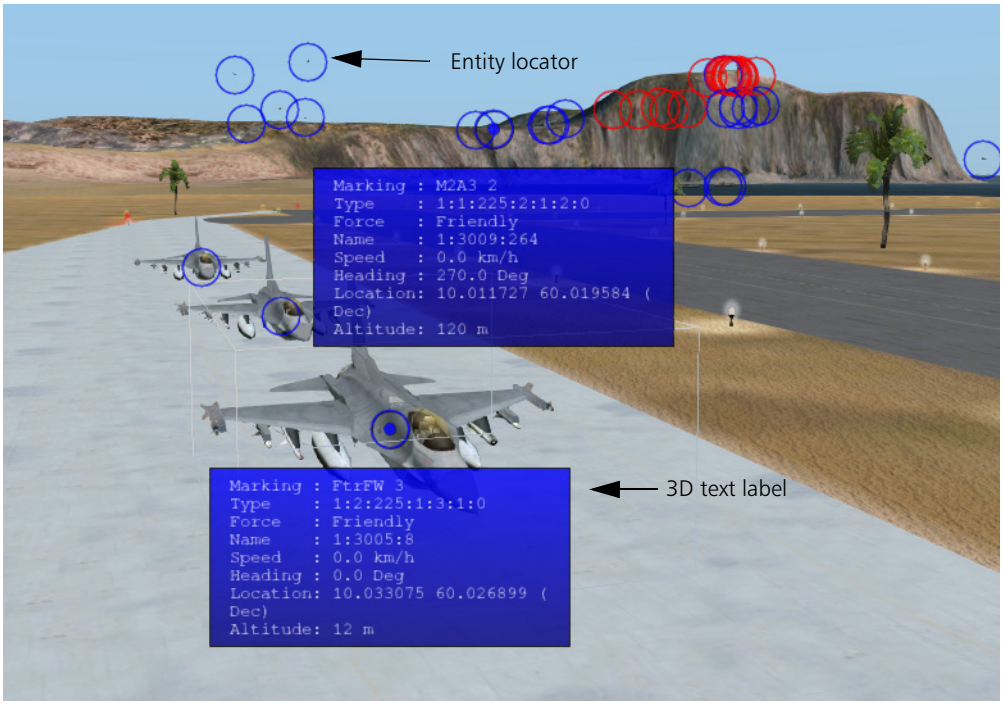


Figure 7-1. Entity labels for 3D models

For 2D icons, entity labels are a set of individually enabled text labels, called symbol decorations (Figure 7-2). Entity labels for 2D icons are supported both in the 2D view and the 3D view. If you resize 2D icons, the labels also get resized. 2D icons can display the entity's:

- Acceleration.
- Altitude above mean sea level.
- Altitude above ground level.
- Heading Indicator.
- Label. If extended labels exist, each label can be displayed separately.
- Location.
- Name.
- Orientation.
- Velocity.
- Weather. Cloud state, precipitation intensity, visibility, wind direction, and wind speed.

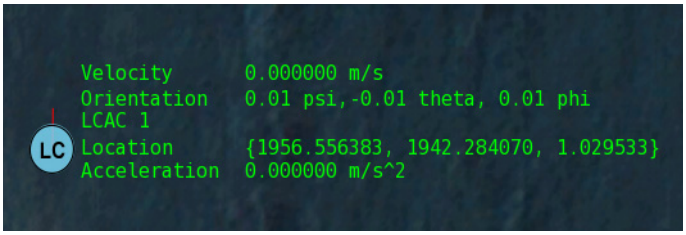


Figure 7-2. Entity labels for 2D icons

If spot reports are enabled, you can display labels for spot reports. For details, please see "Displaying Labels for Spot Reports," on page 13-9, n *VR-Forces Scenario Management Guide*.

7.1.1. Displaying Text Labels for 3D Models

To view a 3D text label, use one of the following methods:

- Move the cursor over the circle that represents the entity.
- On the Display Settings dialog box, Entity Display Settings page, select Show Information Only For Selected Entities (Figure 7-3). Then select the entity whose label you want to see.

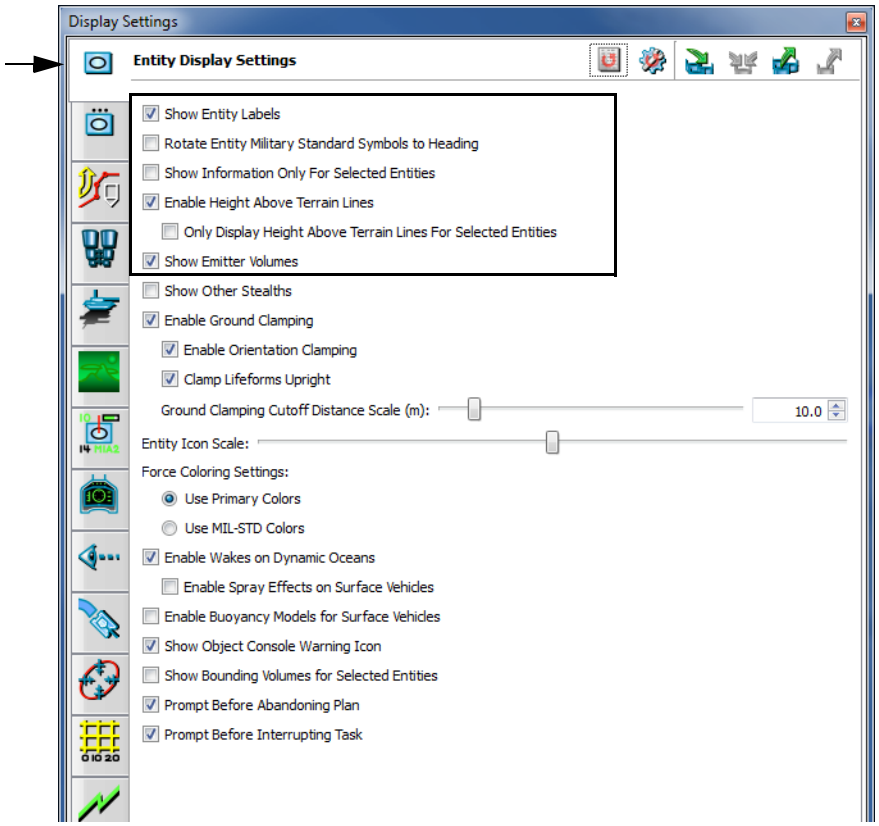


Figure 7-3. Display Settings, Entity Display Settings page

7.1.2. Turning Entity Labels On and Off

To turn the display of entity labels on or off:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 7-3).
3. Select or clear the Show Entity Labels check box.

> You can also enable and disable entity labels by choosing **Settings** → **Entity Labels** or by clicking the Entity Labels button ( ) on the Display Settings Toolbar.

7.1.3. Pinning 3D Text Labels to the Window

You can pin 3D text labels to the window, making them stay visible after you move the cursor away from the entity.

To pin a 3D text label to the window or hide a pinned label:

1. Right-click an entity's locator circle. A menu is displayed.
2. Choose **Toggle Pinned Label**.

Pinning Multiple 3D Text Labels

You can display multiple text labels at one time.

To pin multiple text labels use either of the following methods:

- Right-click and choose **Toggle Pinned Label** for each entity whose label you want to pin.
- On the Display Settings dialog box, Entity Display Settings page, select Show Information Only For Selected Entities (Figure 7-3). Then, in the Objects List Panel, multi-select the entities whose labels you want to see.

7.1.4. Displaying Entity Labels for 2D Icons

You can enable and disable the display of labels for 2D icons by label type. This allows you to control how much screen space is taken up by the labels. You can control the display of entity labels in the Display Settings dialog box or on the Display Settings Toolbar.

To enable and disable display of 2D icon labels:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Symbol Decoration Settings page (Figure 7-4).

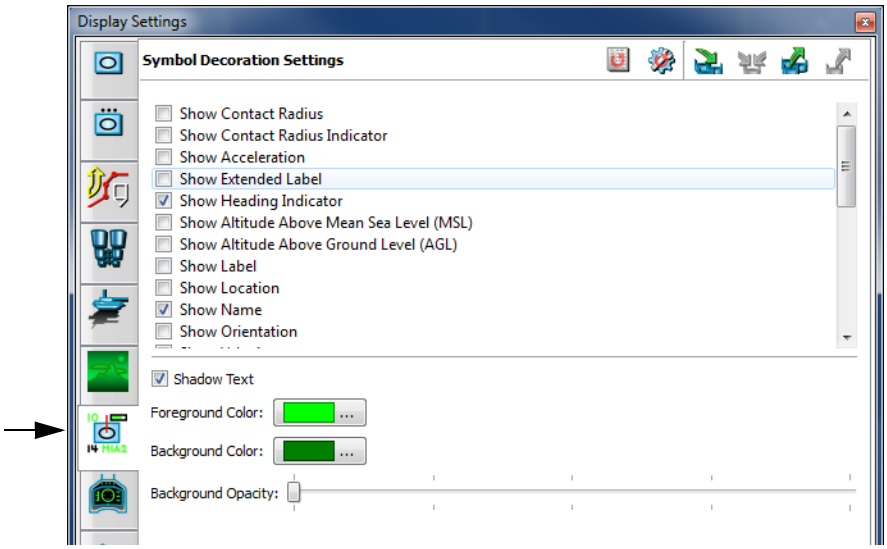


Figure 7-4. Symbol Decoration Settings page

3. Select the check box for each label that you want to display. Clear the check box for each label that you want to hide.

Enabling 2D Labels on the Display Settings Toolbar

The Entity Labels button on the Display Settings Toolbar lets you open a panel that lists the symbol decorations that you can show or hide.

To enable or disable 2D labels on the Display Settings Toolbar:

1. Click the arrow on the Entity Labels button ( ). A small panel is displayed. It is a persistent window.

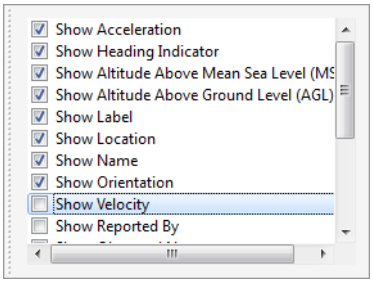


Figure 7-5. Symbol Decoration Panel

2. Select the symbol decorations that you want to display. Clear the ones you want to hide.
3. To hide the Symbol Decorations Panel, click the arrow on the Entity Labels button.

7.1.5. Configuring Shadowing and Background Color for 2D Labels

2D labels are drawn on a background that, by default, is transparent. However, you can increase the opacity of the background to make it visible and can change the background color.

You can add shadowing to the text of 2D labels. This may improve visibility when entities are on light colored terrain.

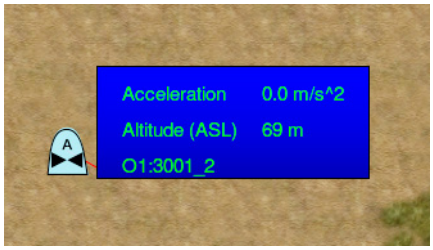


Figure 7-6. 2D label with background

To set the color and opacity of the 2D label background:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Symbol Decoration Settings page (Figure 7-4).
3. To change the opacity of the background, adjust the Background Opacity slider.
4. To change the color of the background:
   a. Click the Background Color color swatch. A color picker opens.
   b. Select the color you want to use for the background.
   c. Click OK.

To configure shadows for 2D labels:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Symbol Decoration Settings page (Figure 7-4).
3. Select or clear the Shadow Text check box.

7.2. Using Extended Labels

By default, one of the ways to identify an entity is by its label. (For more information, please see "Entity Labels," on page 3-14.) Extended labels are additional labels that you can specify on an application-wide basis and define individually for entities. The contents of each label gets sent over the network as part of the entity's information and is displayed in the entity's Information dialog box.

You can add and remove extended labels and change the label name. Changing a label's name does not affect the text that is associated with the label for those entities that use the label.

Extended labels are not scenario-specific. They affect all scenarios. Therefore, if you change a label's name or index, or delete it, be sure that you want this change to apply to all scenarios in which you use the extended label.

Figure 7-7 shows an extended label (called Extended Label) added in the Application Settings dialog box, Session Options page. Figure 7-8 shows the text for the extended label for a specific entity entered in the Edit dialog box and then displayed on the screen.

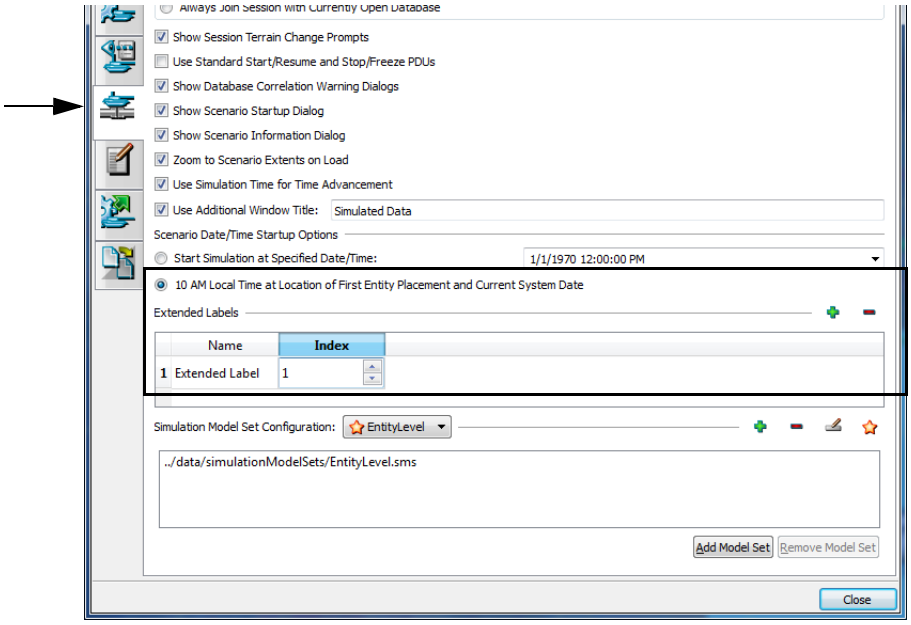


Figure 7-7. Extended label added to session

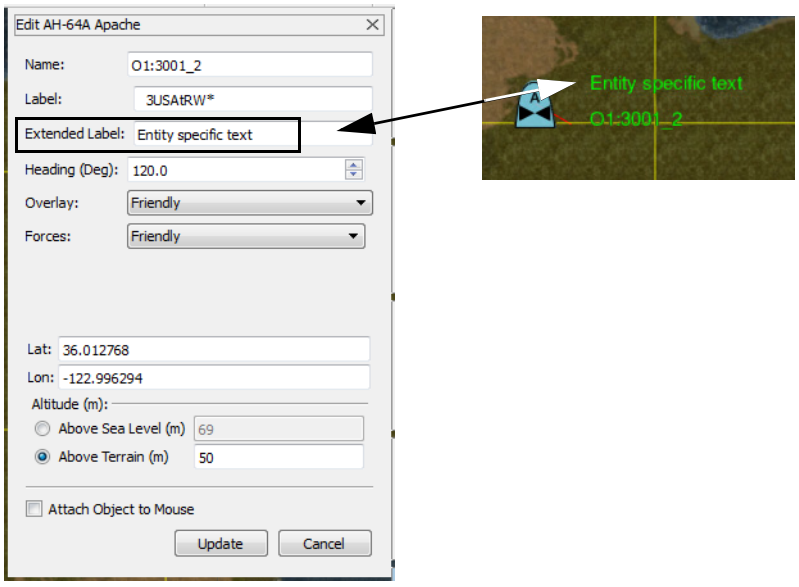


Figure 7-8. Extended label text for an entity

You can display extended labels in Plan View mode the same as other symbol decorations. For more information, please see "Displaying Entity Labels for 2D Icons," on page 7-7, in *VR-Forces Users Guide*.

7.2.1. Adding an Extended Label

To add an extended label:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the Session Options page (Figure 7-7).
3. Click the Add button (+) on the Extended Labels line. A label is added to the list window. It has a generic name (for example, Label 1).
4. Double-click the label name to make it editable.
5. Type the name you want for this label.
6. Edit individual entities to specify the text for this label.

7.2.2. Changing an Extended Label's Index

A label's index is used to store the label text and identify it when sent over the network. In Edit dialog boxes, extended labels are listed in indexed order. You can change an extended label's index. However, while changing an extended label's name does not affect the text that you save for an entity, changing its index could affect existing scenarios. Therefore, use caution in doing so.

To change an extended label's index:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the Session Options page (Figure 7-7).
3. Click the up or down arrow next to a label's index number. The index number will change and, if applicable, the label will move up or down in the list.

7.2.3. Deleting an Extended Label

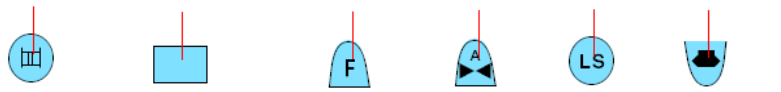
Deleting an extended label affects all scenarios that use that label. (In other words, it is not a scenario-specific setting.) If you delete an extended label, it is immediately removed from all entities that are using that label.

To delete an extended label:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the Session Options page (Figure 7-7).
3. In the Extended Labels window, select the label you want to delete.
4. Click the Delete button (−).

7.3. Entity Icons

The 2D Icons model set uses MIL-STD 2525B icons. Figure 7-9 illustrates basic icon shapes for friendly forces.



Tracked Dismounted infantry Fixed-wing Rotary-wing Surface Subsurface

Figure 7-9. MILSTD-2525B icons

The color and shape code for 2D entity icons is:

- Light blue – friendly.
- Salmon – hostile.
- Light green – neutral.

Symbols have a black border except as follows:

- If there is no border on the top, the entity domain is subsurface, for example, submarines.
- Selected entities have a white border.

Symbols that represent aggregate entities have aggregate identifiers above the symbol, as illustrated in Figure 7-10.

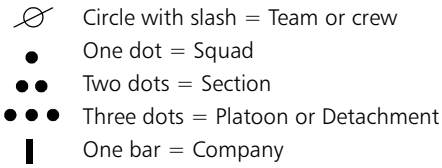


Figure 7-10. Unit echelon indicators

7.3.1. Changing the Size of 2D Icons

The Plan View observer mode uses the 2D Icons model set for entities. You can change the size of 2D entity icons. When many entities are close together, reducing the size of the icons can reduce overlap.

To change the size of entity icons:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 7-11).

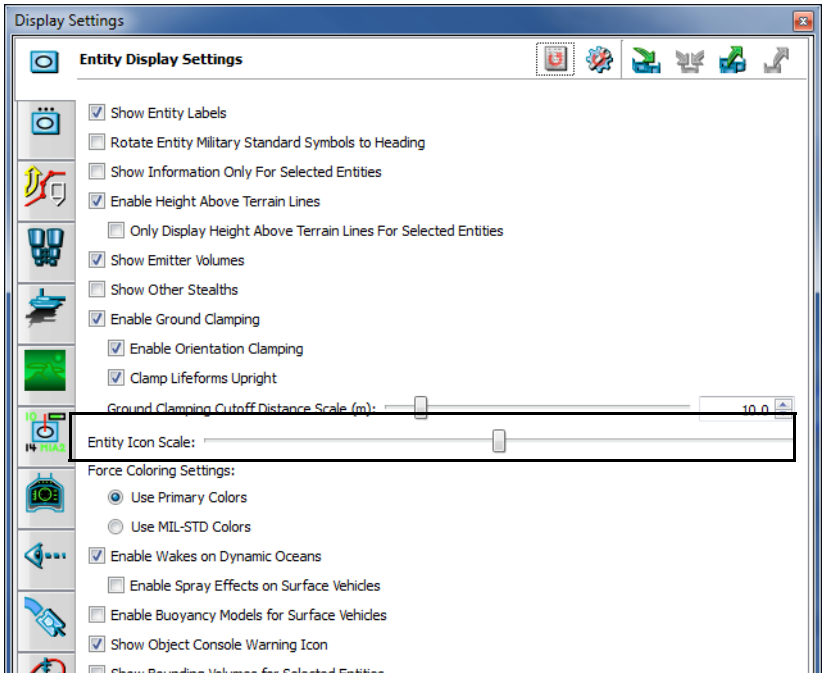


Figure 7-11. Entity Display Settings page

3. Adjust the Entity Icon Scale slider.

You can also change the size of 2D icons from the Display Settings Toolbar.

To change the size of entity icons from the Display Settings toolbar:

1. On the Display Settings Toolbar, click the Icon Scaling button (). The Entity Icon Scale Factor slider opens.
2. Adjust the slider to change the size of the icons.



The Entity Icon Scale Factor slider is a dockable widget that you can undock and leave visible as long as icon scaling is enabled.

7.3.2. Rotating 2D Icons to an Entity's Heading

By default, when an entity's heading changes, the heading indicator points to the new heading, but the icon remains oriented towards zero degrees (North). You can specify that icons rotate to the heading of the entity.



Figure 7-12. 2D icon rotated to heading

To rotate icons to the entity's heading:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 7-11).
3. Select the Rotate Entity Military Standard Symbols to Heading check box.

7.4. Model Sets

VR-Forces includes the following model sets:

- 3D Models. Realistic, 3D, textured models with articulated parts.
- 3D Colorized Models. 3D model caricatures with minimal articulation. These models do not have textures, they are solid colors, by force. You can change the coloring between primary colors and MIL-STD icon colors.
- 2D icons. MIL-STD 2525B icons.

You can use any of the model sets in both 3D and 2D views. In the 2D view, the 3D models can only be viewed from the top.

7.4.1. Changing the Model Set

Each observer mode specifies a model set. You can change the model set associated with an observer mode by editing it (for details, please see "Editing an Observer Mode," on page 10-4), but you can also dynamically change the model set being used.

➤ To change the model set, choose **Observer** → **Set Model Set** → *model_set*, where *model_set* is one of the options on the menu, or select a model set from the Set Model Set button ( ) on the Observer Settings Toolbar. (The button changes to show the currently selected model set.)

7.5. Exaggerating the Scale of 3D Entity Models

If you view the terrain in Stealth observer mode, beyond a certain distance most 3D entity models are too small to see. Model scaling lets you exaggerate the scale of entity models to see them at a great distance. Figure 7-13 shows the same scene with model scaling off and on. When model scaling is off, only the largest entities are visible. When models are scaled, they are all visible.

VR-Forces supports the following types of model scaling:

- XR scaling. XR scaling tries to give all entities approximately equal visibility. It does not preserve size relationships between entities. The visual effect of model scaling depends on an entity's size and the observer's distance from the entity. When the observer is close to an entity, scaling models may not make any noticeable difference in its size.
- Linear scaling. Linear scaling scales all entities by the current model scale factor. The distance from the observer is not taken into account. A model scale factor of 10 scales all entities to be 10 times normal size.

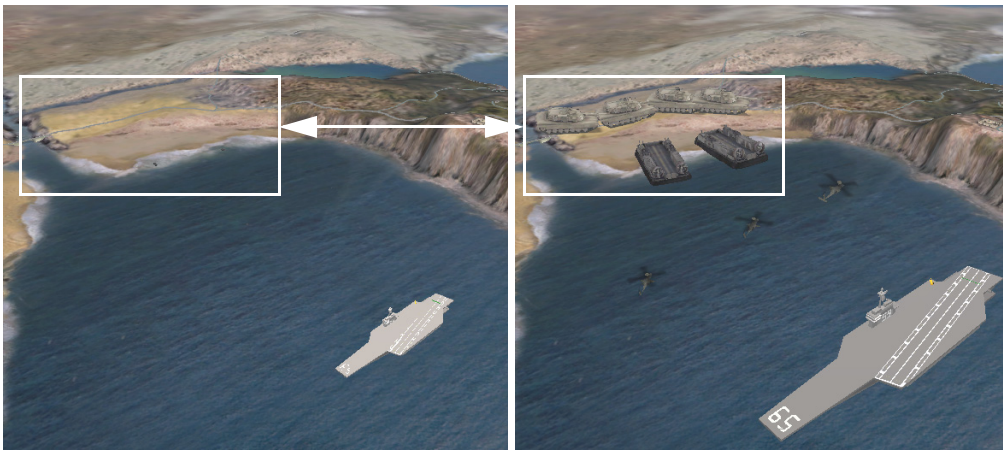


Figure 7-13. Model scaling off and on (Stealth observer mode)

Another way to quickly change the visibility of entities is to switch to XR observer mode. In XR mode, entities are immediately scaled and are represented by colorized 3D models to enhance their visibility.



Figure 7-14. XR mode

To scale entity models:

1. Choose **Observer** → **Advanced Settings**. The Display Settings dialog box opens to the Observer Settings page (Figure 7-15).

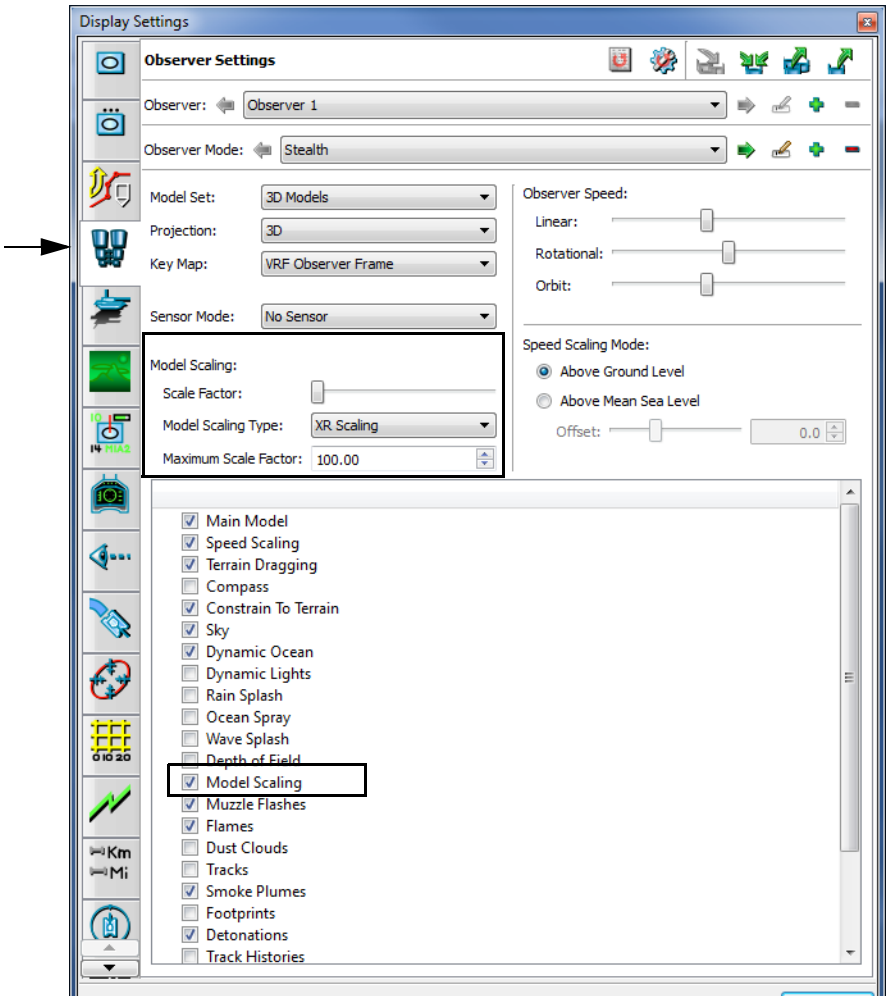


Figure 7-15. Observer Settings page

2. In the list of observer attributes, select Model Scaling.
3. In the Model Scaling Type list, select the type of scaling you want to use – XR or linear.
4. Optionally, change the Maximum Scale Factor.
5. Adjust the Scale Factor slider to the amount of scaling that you want.

You can also scale models from the Observer Settings Toolbar.

To scale models from the Observer Settings Toolbar:

1. If model scaling is not enabled, click the Model Scaling button ( ) on the Observer Settings Toolbar.
2. Click the down arrow on the button. The Scale Factor slider is displayed (Figure 7-16).



Figure 7-16. Scale Factor slider

3. Adjust the slider to scale the models.



The Scale Factor slider is a dockable widget that you can undock and leave visible as long as model scaling is enabled.

7.6. Hiding 3D Models

You can hide 3D models and 3D colorized models.

To hide 3D models:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 7-15).
3. In the list of observer settings, clear the Main Model check box.

7.7. Changing Force Colors

By default, entity labels, detonation lines, and other force-related entity graphics (including the 3D Colorized Model model set) are displayed using primary colors (red, blue, and green). You can change the force coloring to use MIL-STD 2525B colors (light blue, light green, and salmon).

To change force coloring:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 7-11).
3. Select the Force Coloring Settings option that you want.

7.8. Displaying HAT Lines for Entities (3D Only)

Height-above-terrain (HAT) lines are vertical lines that run from an entity to the ground and display the altitude of the entity (using whatever units are configured on the Display Units Settings page) (Figure 7-17). They help you understand the relationship of the entity to the terrain.

HAT lines are shown only for entities that could be above or below the ground, such as fixed and rotary-wing vehicles. You can show and hide these lines globally and you can specify that they be shown only for the selected entities.

When you create or edit an entity, VR-Forces displays HAT lines even if they are otherwise disabled.

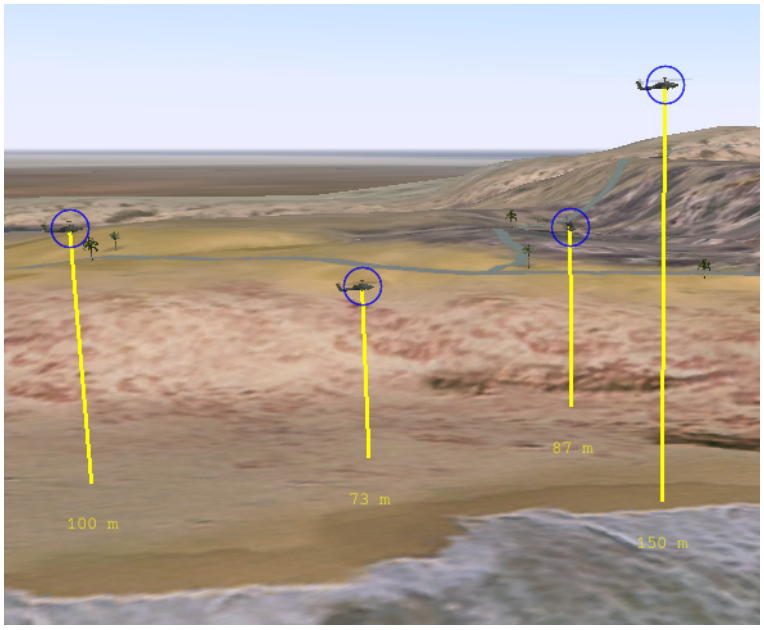


Figure 7-17. Entity height-above-terrain lines

! Displaying height above terrain lines can reduce the frame rate.

To show or hide height-above-terrain lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 7-4).
3. To display lines, select the Enable Height Above Terrain Lines check box.
4. To display lines only for selected entities, select the Only Display Height Above Terrain Lines For Selected Entities check box.

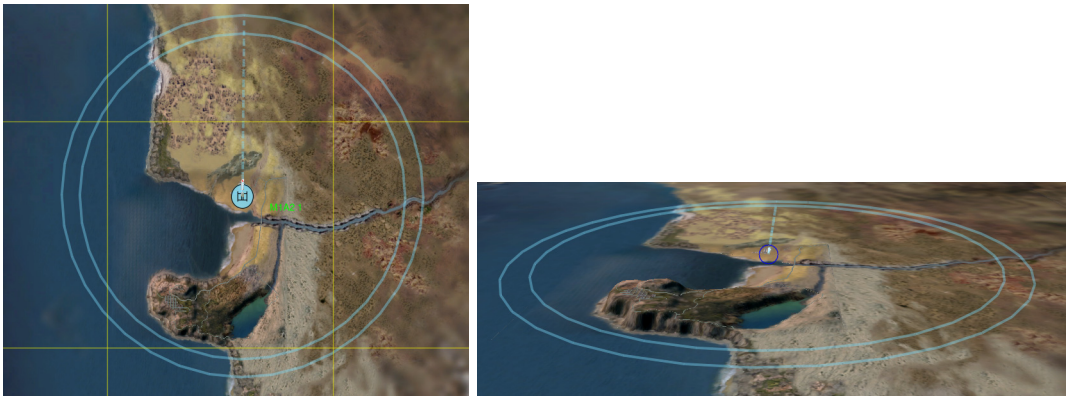


You can also enable and disable entity HAT lines by choosing **Settings** → **Entity Height Above Terrain Lines** or by clicking the Entity Height Above Terrain Lines button () on the Display Settings Toolbar.

7.9. Displaying Threat and Sensor Range Rings

You can display an entity's threat range – the area within which its armaments are effective. The threat range is displayed as a circle, with a line indicating the armament and its direction of fire (Figure 7-18). A ring is displayed for each weapon system that the entity has. For example, in Figure 7-18, the M1A2 entity has an inner ring for its 62mm machine gun and an outer ring for its 120mm main gun.

You can also display range rings for an entity's sensors (Figure 7-19).



2D 3D

Figure 7-18. Range rings (entity-level scenario)

Entities in aggregate-level scenarios display range rings that show their combat effectiveness and a footprint that represents the physical area that the aggregate occupies (Figure 7-19). Aggregate range rings have breaks to show the front, rear, left flank, and right flank sectors.

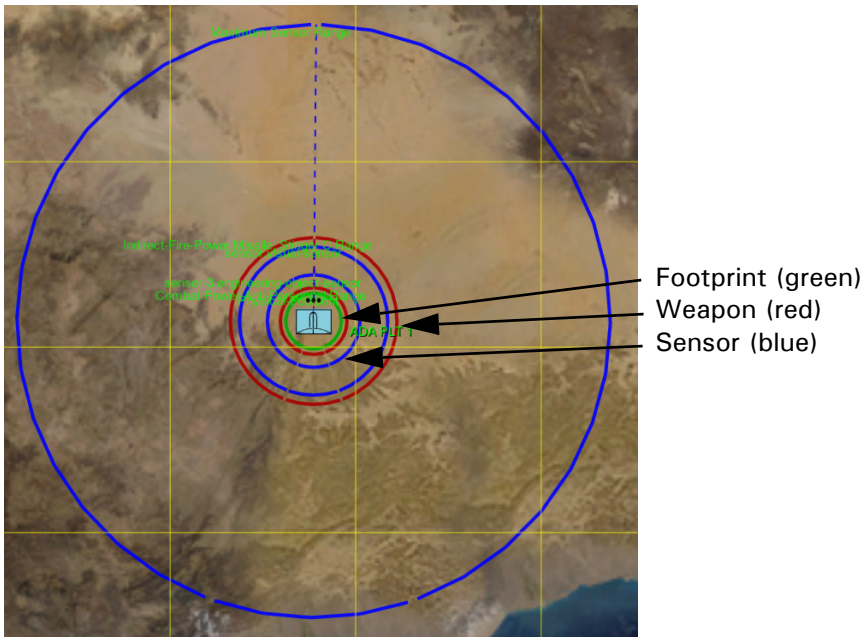


Figure 7-19. Aggregate-level range rings

You enable range rings globally. They are an observer-specific option. You display range rings on a per-entity basis. You can globally configure the color of rings, their transparency, and the display of labels. In 3D views, you can specify whether or not to display a range sphere (Figure 7-20).

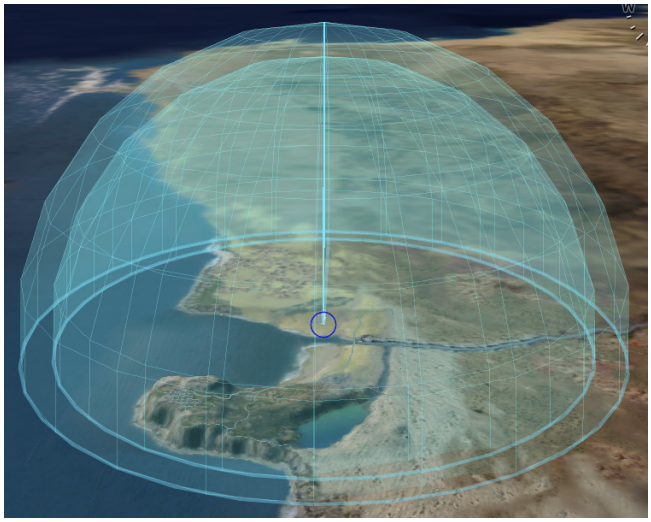


Figure 7-20. Range ring sphere



On small databases, an entity's range ring may extend beyond the edge of the terrain display.

7.9.1. Enabling and Disabling the Display of Range Rings

You can enable or disable the display of range rings on the Observer Settings page.

To enable or disable the display of range rings:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 7-15).
3. In the list of options, select or clear the Range Rings check box.



You can also enable and disable range rings by choosing **Observer** → **Range Rings** or by clicking the Range Rings button () on the Display Settings Toolbar.

7.9.2. Displaying Range Rings for an Entity

You display range rings on a per-entity basis.

To display or hide range rings for an entity:

1. Enable range rings, as described in "Enabling and Disabling the Display of Range Rings," on page 7-23.
2. Select an entity.
3. Choose **Entities** → **Display Range Rings**.

7.9.3. Configuring the Display of Range Rings

You can configure the color of range rings and their transparency by weapon type and sensor type. You can also configure the display of range ring labels (Figure 7-21). A label can display the entity name, weapon type, and range of the weapon.

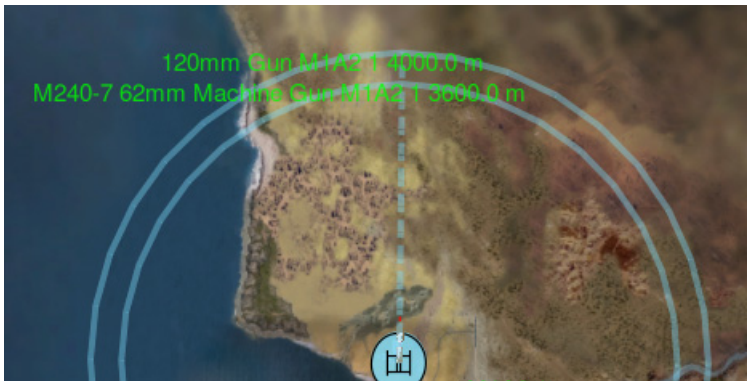


Figure 7-21. Range ring labels

To configure the display of range rings:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Range Rings Settings tab (Figure 7-22).
3. To display a sphere in 3D views, select the Show Range Sphere in 3D Using Opacity Modifier check box. Optionally, change the opacity modifier value.
4. To display the type of sensor or weapon system, select the Show Weapon Name check box.
5. To display the name of the entity, select the Show Entity Name check box.
6. To show the range of the sensor or weapon system, select the Show Range Value check box.
7. To automatically display range rings when an entity is created or discovered on the network, select Display Range Rings by Default.

> If this setting is selected, when you load a scenario, range rings are displayed for all entities.

8. To display range rings for a selected entity that has not enabled range rings, select Display Range Rings for Selected.

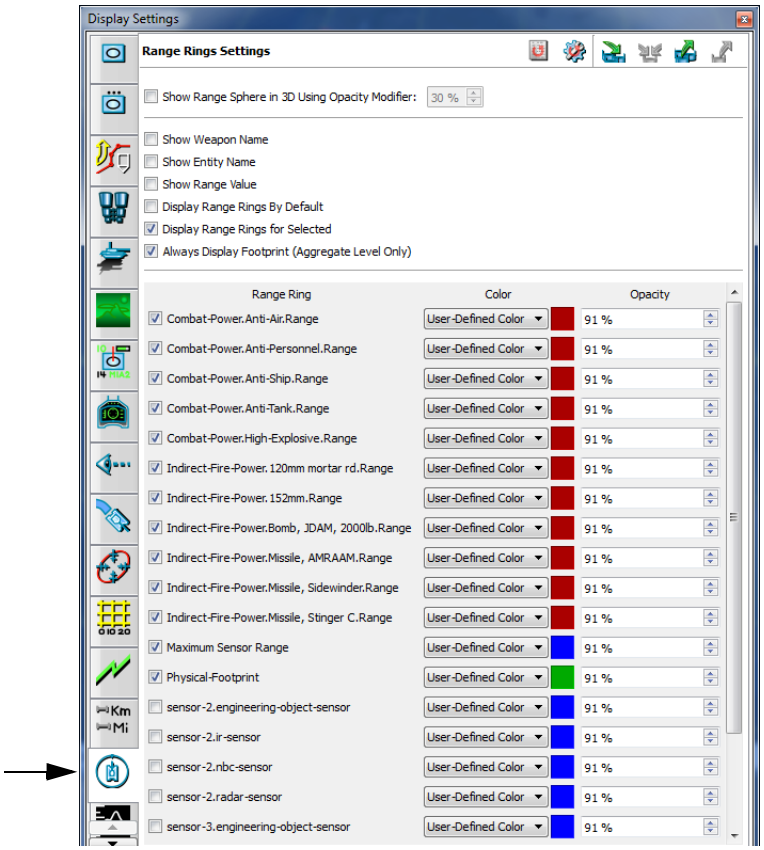


Figure 7-22. Range Rings Settings page (*AggregateLevel.sms*)

9. To always show the footprint of an entity, select Always Display Footprint.
10. To enable or disable display of range rings for a particular weapon system, select or clear the check box for that system.
11. To change the color used to display a range ring:
    a. For the type of weapon system or sensor whose color you want to change, select User-Defined Color in the Color column list. The color swatch next to the list changes.
    b. Click the color swatch. A color picker dialog box opens.
    c. Select the color you want to use.
    d. Click OK.
12. To change the opacity for a range ring, for the weapon system whose range ring opacity you want to change, change the value in the Opacity column.

7.10. Displaying Entity Resources

VR-Forces tracks the amount of resources that an entity has. You can view the status of resources in the Entity Information dialog box.

7.10.1. Viewing Numerical Resource Data

The Entity Information dialog box lists the entity's current amount of resources, its starting amount, and the percentage remaining (Figure 7-23).

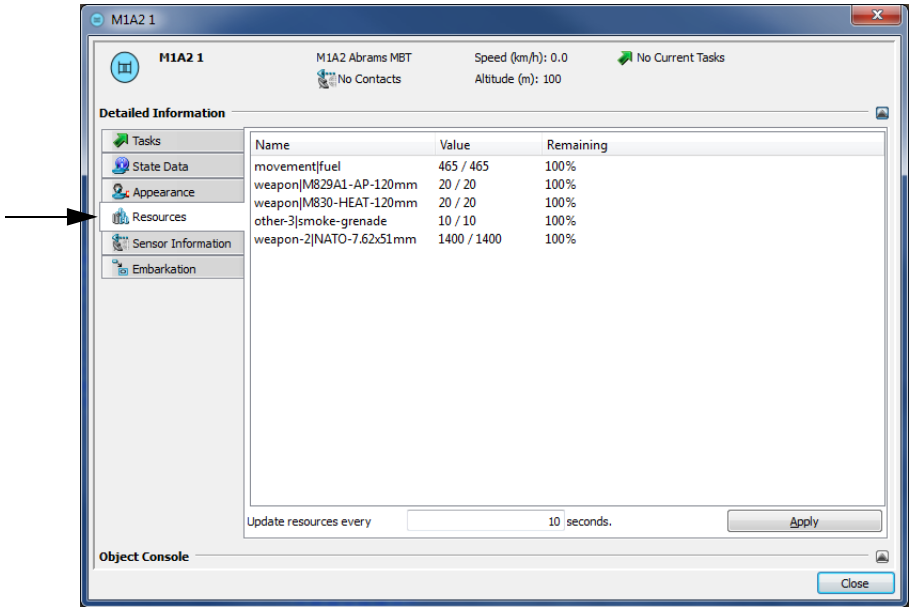
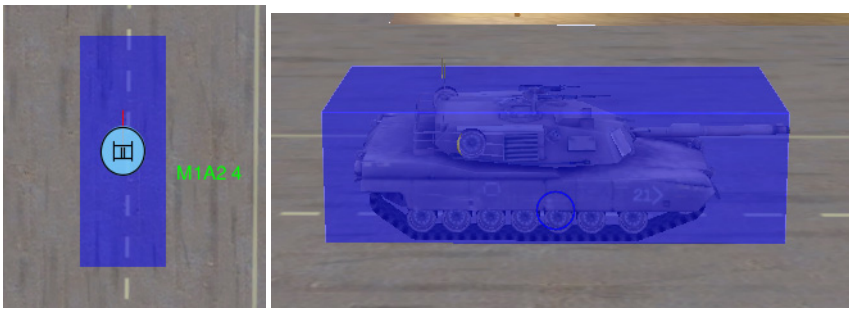


Figure 7-23. Resource Information page

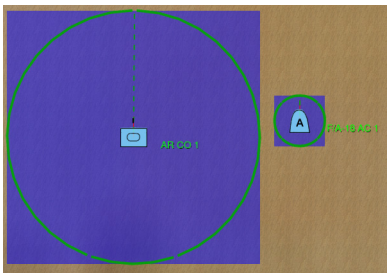
➤ To view entity resource status, on the Entity Information dialog box, select the Resource Information page.

7.11. Displaying Bounding Volumes

You can display entity bounding volumes. Bounding volumes show the area around an entity that the back-end uses for intersection calculations. Bounding volumes are shown as translucent volumes in 3D and translucent rectangles in 2D (Figure 7-24). For entities in aggregate-leel scenarios, the bounding volume is equivalent to the footprint.



2D 3D



Aggregate-level bounding volumes

Figure 7-24. Bounding volumes

To enable or disable bounding volumes:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 7-15).
3. Select the observer mode for which you want to enable or disable bounding volumes.
4. Select or clear the Bounding Volume check box.

7.11.1. Showing Bounding Volumes Only for Selected Entities

You can configure VR-Forces so that it shows bounding volumes only for selected entities.

To display bounding volumes only for selected entities:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 7-11).
3. Select the observer mode for which you want to enable bounding volumes.
4. Select the Bounding Volume check box.
5. Select the Entity Display Settings page (Figure 7-3).
6. Select the Show Bounding Volumes for Selected Entities check box.

7.12. Configuring Object Console Messages

The Information dialog boxes for entities and tactical graphics have a console that displays messages sent from the simulation engine, from an entity's plan, from other entities, and from scripted tasks. You can set the notification level that controls the display of these messages. You can save them to the clipboard, and you can clear the display. These messages are also sent to the Object Console Summary Panel.

Scripted tasks can ask questions. You can answer these questions from the Object Console Summary Panel.

7.12.1. Setting the Notification Level for Console Messages

Console messages are displayed only if they match or exceed the notification level set for that object. The notification level is set individually for each object. You can set the notification level in the object's Information dialog box, with the Notify Level set data request (for entities), or with the **objectConsoleNotifyLevel** parameter in *vrfSim.mtl*. (For the set data request procedure, please see "Notify Level," on page 10-31, in *VR-Forces Scenario Management Guide*.)

To set the notification level for an object's console in the Information dialog box:

1. Select the entity.
2. Choose **Entities** → **Information**. The Information dialog box opens.
3. Expand the Object Console section (Figure 7-25).

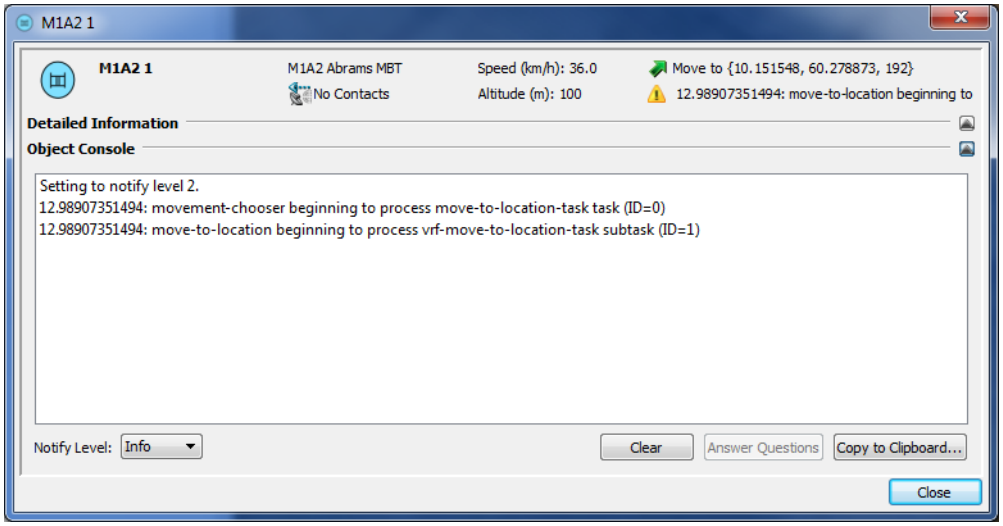


Figure 7-25. Entity Information dialog box, Object Console

4. Select a notification level from the Notify Level list.

7.12.2. Viewing All Console Messages

You can view all object console messages in one place – the Object Console Summary Panel. Messages are displayed based on the notification levels of the individual entities in the scenario. You can set the notification level in an entity's Information dialog box or using the Notify Level set data request.

When VR-Forces sends a message to an entity's console, it displays an exclamation point next to the entity until you view the message (unless you disable this feature). The messages in the Object Console Summary Panel also display an exclamation point until you open the Information dialog box for that entity or click Acknowledge All.

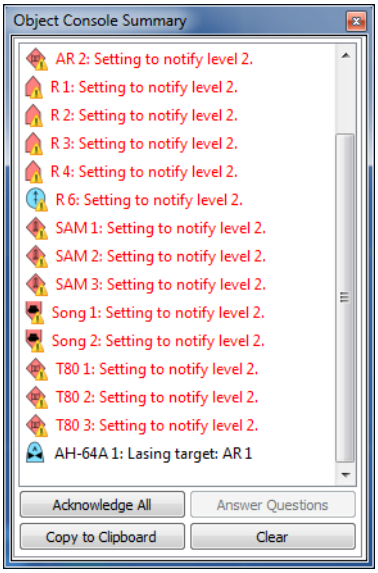


Figure 7-26. Object Console Summary Panel

➤ To display the Object Console Summary panel, choose **View** → **Object Console Summary Panel**.

➤ To remove the exclamation point from messages in the Object Console Summary Panel, click Acknowledge All.

7.12.3. Answering Questions from Scripted Tasks

Scripted tasks can send questions to the object console. Questions are indicated by a question mark next to the message . The script will offer a set of responses as buttons, option buttons, or a list. You can answer questions from the Object Console and the Object Console Summary Panel.

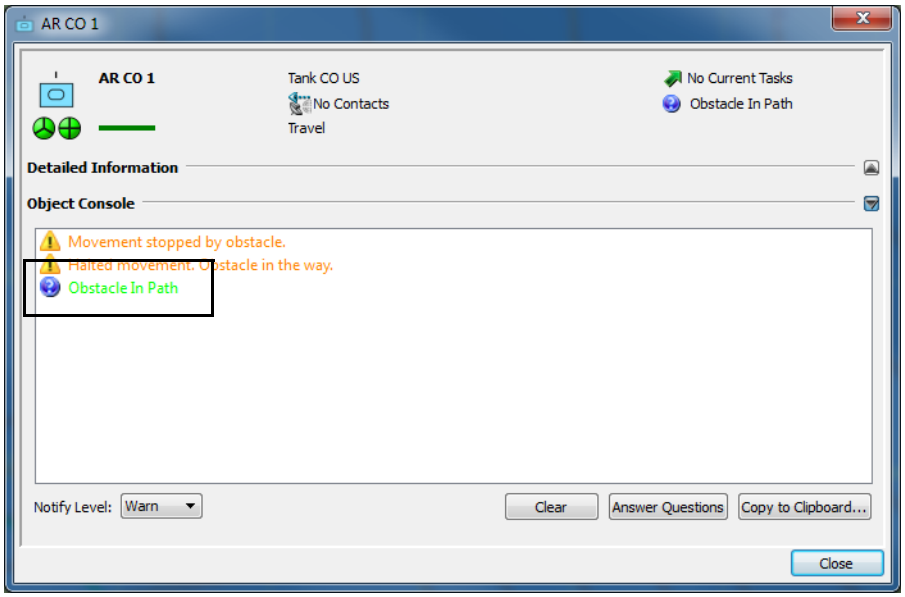


Figure 7-27. Question in console

To answer a question from a scripted task:

1. On the Object Console of Object Console Summary Panel, click Answer Questions. A dialog box with the possible answers is displayed.
2. Choose an option.
3. Click OK.

7.12.4. Saving Console Messages

If you want to save the messages that are sent to the console, you can copy them to the clipboard and then paste them into a file.

To save console messages to the clipboard:

1. Select the object.
2. Choose **Entities** → **Information**. The Information dialog box opens (Figure 7-25).
3. To save all messages in the console, click Copy to Clipboard. To save specific messages, select them and click Copy to Clipboard.
4. Paste the text into a the text editor or word processor of your choice.

7.12.5. Displaying an Object Console Warning Icon

VR-Forces can display a notification icon when an object receives a message on its object console (Figure 7-28). This option is enabled by default, however the default notification level is Warn and objects do not typically get messages at this level. The more likely use for this icon is to set the notification level to Info or lower and then check the console when a notification is received. For details about how to set the notification level, please see "Setting the Notification Level for Console Messages," on page 7-29.



Figure 7-28. Object console warning icon

The icon is not displayed if the Information dialog box is open. If an icon is displayed for an object, when you open its Information dialog box, the icon is removed.

To enable or disable the console warning icon:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 7-3).
3. Select or clear the Show Object Console Warning Icon check box.

7.12.6. Clearing the Object Console

➤ To clear the messages from the object console, click Clear.



8. Displaying Entity Effects

This chapter describes how to manage special visual effects that can be applied to entities and the VR-Forces scene.

Entity Effects in Aggregate-Level Scenarios 8-3
Using Ground Clamping 8-3
Configuring Ground Clamping 8-4
Trajectory Smoothing 8-5
Configuring Trajectory Smoothing 8-6
Inset Views 8-7
Displaying Trailing and Decal Effects (3D Only) 8-9
Specifying Models for Trailing Effects and Decal Effects 8-11
Displaying Track Histories 8-11
Configuring Track Histories 8-12
Displaying a Terrain Profile for a Track History 8-13
Viewing Fire and Detonation Effects 8-14
Viewing Fire and Detonation Lines 8-14
Viewing Aggregate Combat Graphics 8-15
Displaying Laser Designators 8-17
Displaying Supply Lines 8-18
Displaying Tactical Smoke 8-18
Displaying Electromagnetic Emissions 8-19
Displaying Radio Communication Lines 8-20
Enabling Radio Communications Graphics 8-21
Configuring Radio Communication Lines Lifetime and Display Mode 8-22
Configuring the Color of Radio Communication Lines 8-23

Displaying Cockpit Displays (3D Only) 8-25
Displaying Instrument Panels 8-27
    Configuring Entities to use Instrument Panels..... 8-28
Displaying Shadows (3D Only) 8-29
    Configuring Shadows..... 8-31
Using the Depth of Field Effect 8-32
    Configuring Depth of Field..... 8-34
Displaying Wakes and Spray Effects..... 8-35
Enabling Buoyancy for Surface Entities..... 8-36

8.1. Entity Effects in Aggregate-Level Scenarios

Many of the entity effects described in this chapter apply to individual entities in Stealth mode. By default, entities in aggregate-level scenarios do not use 3D models. Therefore, entity effects such as trailing effects, smoke clouds, and fire and detonation effects are not supported.

8.2. Using Ground Clamping

Because of differences in terrain databases among exercise participants, entities can sometimes appear to be underground or hovering above the terrain surface. To compensate, VR-Forces can clamp land and sea entities to the terrain surface. When ground clamping is enabled, VR-Forces keeps an entity anchored to the terrain surface regardless of the altitude data contained in its state update.



Use of ground clamping can reduce your frame rate. If you correlate the simulation and visualization terrain databases, you can eliminate the need to use ground clamping.

To enable or disable ground clamping:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 8-1).

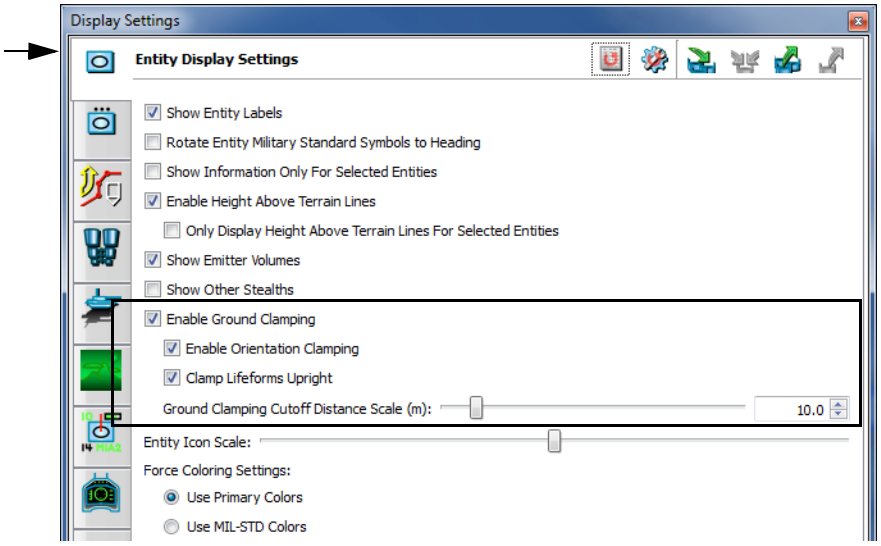


Figure 8-1. Ground clamping settings

3. Select or clear the Enable Ground Clamping check box.



You can also enable and disable ground clamping by choosing **Settings** → **Ground Clamping**, or clicking the Ground Clamping button (▾) on the Display Settings Toolbar.

8.2.1. Configuring Ground Clamping

You can configure the following aspects of ground clamping:

- Enable or disable ground clamping.
- Enable or disable orientation clamping.
- Force lifeforms to remain upright relative to the plane of their location when moving up slopes or steps, rather than being perpendicular to the terrain.
- The distance from the terrain within which entities are clamped.

To configure ground clamping:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 8-1).
3. To enable or disable a ground clamping option, select or clear its check box.
4. To specify the distance from the terrain in which entities are ground clamped, adjust the Ground Clamping Cutoff Distance Scale slider or type a value, in meters, in the box. You can also access the Ground Clamping Cutoff Distance Scale slider by clicking the down arrow next to the Ground Clamping button (▾) on the Display Settings Toolbar.



The setting for the Ground Clamping Cutoff Distance Scale maximizes at 100 meters. If you set it to the maximum, the distance is infinite and all entities are ground-clamped.

8.3. Trajectory Smoothing

From the time VR-Forces receives a state update message for an entity, until it receives the next update, it calculates the location of the entity based on the heading and acceleration in the most recent state update. This process is called dead-reckoning. When the next state update arrives for the entity, VR-Forces moves it to the new position. If the actual position is significantly different from the dead-reckoned position, the entity could appear to jump from the dead-reckoned position to the actual position. VR-Forces can eliminate these discontinuities by smoothing entity trajectories and rotation paths.

Figure 8-2 illustrates dead-reckoning and smoothing.

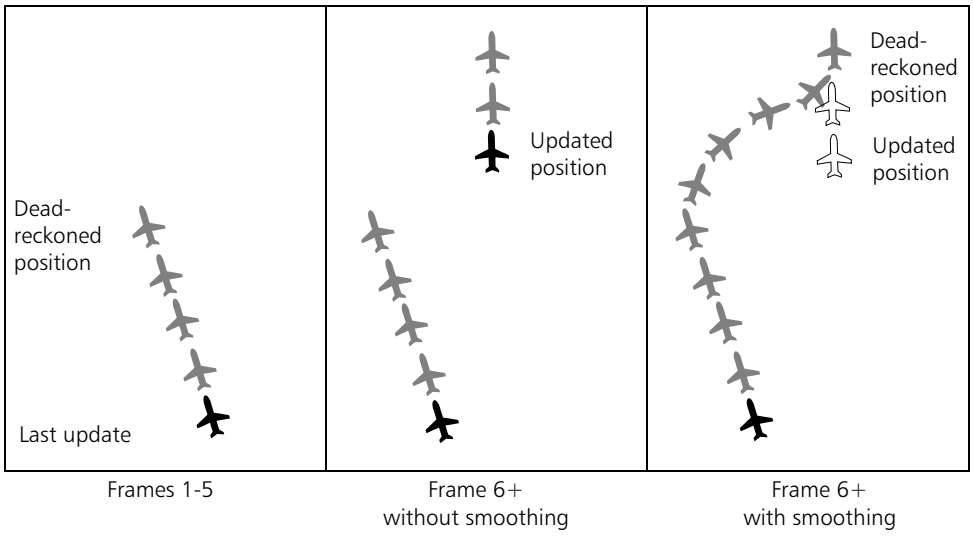


Figure 8-2. Dead-reckoning and smoothing

The smooth period is the period of time over which smoothing takes place. The default period is 1/10 second. If course corrections appear abrupt with this setting, you may want to increase the smooth period.



Smoothing makes fast-moving entities, such as fixed wing aircraft, look better. Slow moving entities, such as dismounted infantry, may look worse. If your simulation is mostly one or the other of these types of entities, enable or disable smoothing as appropriate for the best visualization experience.

8.3.1. Configuring Trajectory Smoothing

To configure trajectory smoothing:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the Performance Options page (Figure 8-3).

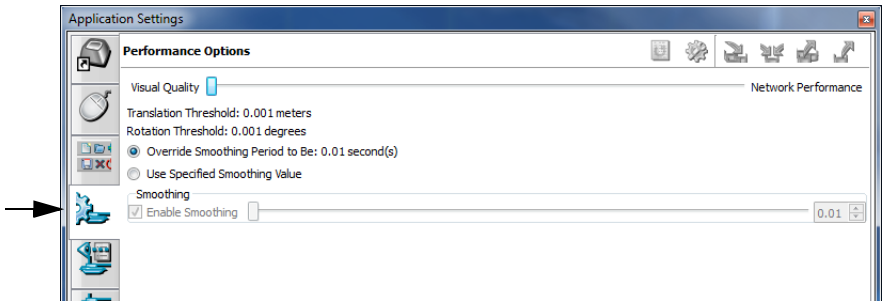


Figure 8-3. Performance Options page

3. Select the Use Specified Smoothing Value option. The Enable Smoothing check box becomes enabled.
4. To enable or disable smoothing, select or clear the Enable Smoothing check box.
5. To change the smooth period, adjust the Smoothing Period slider or type a value, in seconds, in the box.

8.4. Inset Views

You can display inset views of individual entities. Figure 8-4 shows a helicopter hovering over the warehouse area in the Ala Moana inset in the VR-TheWorld Online - MAK Earth terrain with an inset view of a car driving through the town.



Figure 8-4. Inset view

! By default, when you create an inset view, the observer in the inset view is attached to the entity in Mimic mode. This has the side effect of causing the 3D model in the main window to disappear. (The reverse occurs if you attach the observer in the main window to the entity in Mimic mode.) To make the model reappear, select the inset view and change the attach mode or detach the observer.

You can change the observer position and orientation in an inset view using keyboard controls. You can also edit window and channel properties for the window in the Display Engine Configuration Editor.

When you create an inset view, the title bar contains the entity's name. However, the window is not actually tied to the entity and you can change the view in the inset view and view other entities. The title bar will continue to display the name of the entity for which the window was created.

To open an inset view of an entity:

1. Right-click the entity (or its entry in the Objects List Panel). A menu opens.
2. On the menu, choose **Create Inset View**.

8.5. Displaying Trailing and Decal Effects (3D Only)

VR-Forces supports trailing effects such as contrails for missiles and aircraft and dust clouds for vehicles. It also supports decal effects, such as tire tracks, footprints, and wakes. In essence, trailing effects are effects that are dispersed through the air, while decal effects are effects that are printed on the ground.

Trailing effects and decal effects are properties of an observer mode. You can enable and disable them on a per-observer basis.

Figure 8-5 illustrates some trailing effects.



Figure 8-5. Tank tracks, dust cloud

To enable or disable trailing effects or decal effects:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 8-6).

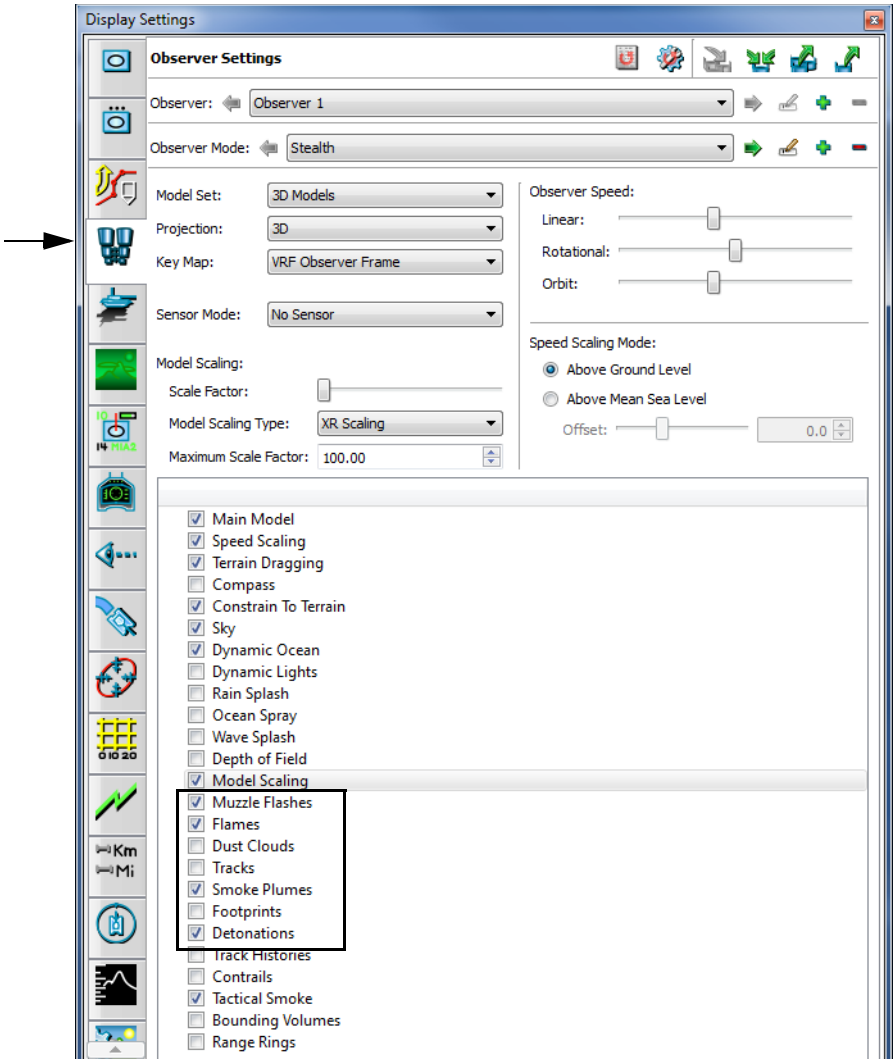


Figure 8-6. Display Settings dialog box, Observer Settings

3. Select or clear the check box for each effect that you want to change.



You can also enable and disable trailing and decal effects by choosing **Observer** → *effect* or clicking the appropriate button on the Observer Settings Toolbar.

8.5.1. Specifying Models for Trailing Effects and Decal Effects

Trailing effects and decal effects are configured through model definitions. (For the purposes of configuration, trailing effects are categorized as decals.) Each effect must have a model definition that uses a decal schema. Then, in an entity's entity definition, you configure a visualizer for the effect. For details about creating model definitions, please see Section 17.3, "Creating and Editing Model Definitions," in *VR-Forces Configuration Guide*.

8.6. Displaying Track Histories

Track histories allow you to track the paths of entities during an exercise, including the path of munitions as they travel from the shooter to the target. When track histories are enabled, each entity leaves a track ribbon. The ribbon is colored based on the entity's force. Figure 8-7 illustrates track histories.

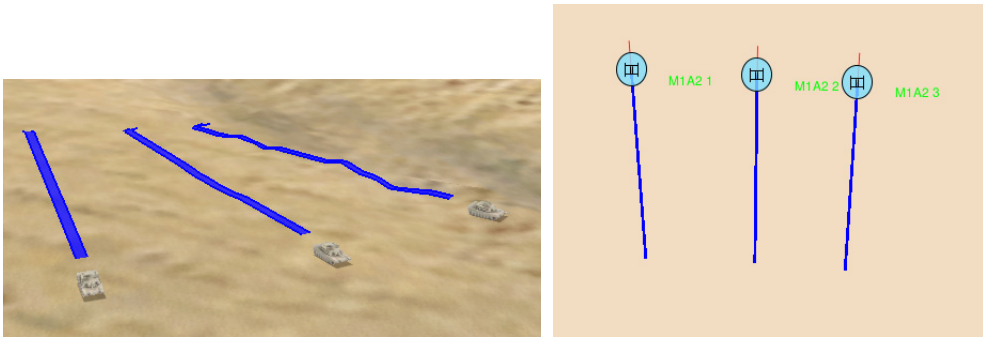


Figure 8-7. Track history

Track histories are a property of an observer mode. You can enable and disable them on a per-observer basis.



Entity track histories have a significant performance and memory cost. For simulations with large entity counts, turning track histories off can improve performance and memory usage.

To enable or disable display of all track histories:

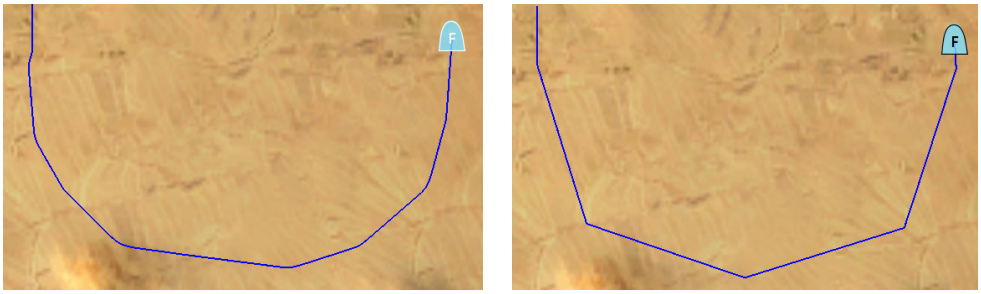
1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 8-6).
3. Select the observer mode for which you want to change the Track Histories setting.
4. Select or clear the Track Histories check box.



You can also enable or disable track histories by choosing **Observer** → **Track Histories** or clicking the Track Histories button (🛰) on the Observer Settings Toolbar.

8.6.1. Configuring Track Histories

You can configure how long track history ribbons are drawn. A track history is composed of a series of segments, joined together to form a line. You can configure how long each segment is and how many segments at a time can be displayed. Drawing shorter segments improves the accuracy of the track history, but can degrade performance. Increasing the maximum number of segments provides a longer history, but can clutter up the screen if there are many entities and can also affect performance. Figure 8-8 illustrates the visual result of drastically changing the segment length. The left track history uses the default segment length of 0.5 meters. The right track history has a segment length of 5000 meters.



0.5 M segment length

5000 M segment length

Figure 8-8. Track history segment length

When a track history reaches the maximum number of segments configured, it starts dropping segments from the beginning of the history (the oldest segments).

You can specify that track histories have an infinite length. In this case they never drop segments.

If you specify infinite length for track histories, it is possible that you could eventually use up all the memory in your computer and the program would crash.

To configure track histories:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Track History Settings page (Figure 8-9).

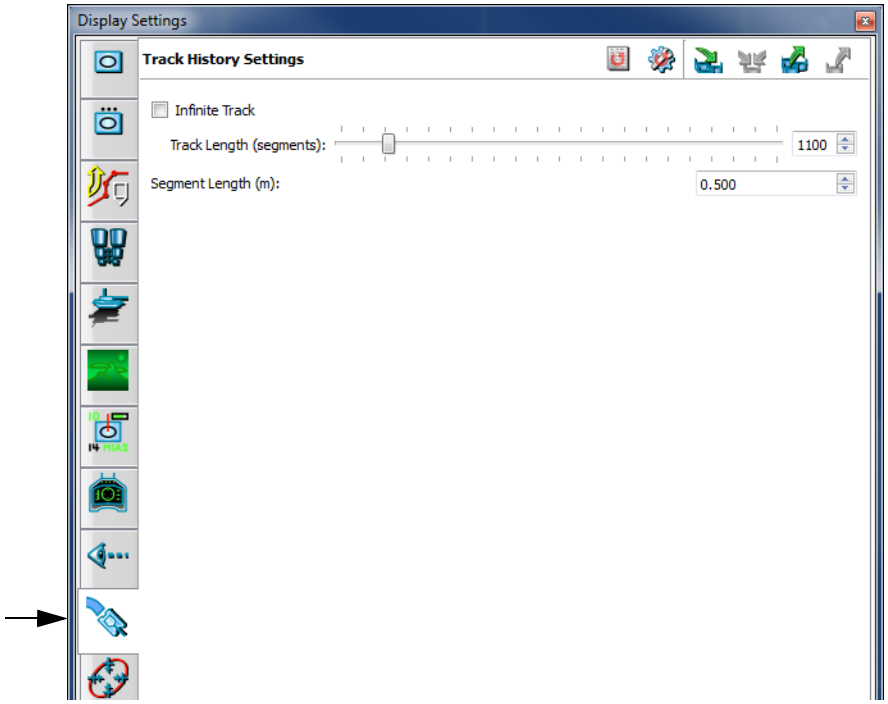


Figure 8-9. Track History Settings page

3. If you want track histories to have an indefinite length, select the Infinite Track check box.
4. To change the number of segments that can be displayed for a track history, adjust the Track Length slider, or enter a value in the spin box.
5. To specify the length of each track history segment, enter a value in the Segment Length box.

8.6.2. Displaying a Terrain Profile for a Track History

You can display a terrain profile for an entity track history. (For complete information about terrain profiles, please see Chapter 13, *Using Terrain Profiles*.)



The track history terrain profile does not display the track history that gets drawn in the window when you enable track histories. It creates an independent track that starts when you open the terrain profile window.

To display a terrain profile for a track history:

1. Select the entity whose track history terrain profile you want to see.
2. Choose **Entities** → **Track History Profile**. The Terrain Profile dialog box opens. It updates as the entity moves.

8.7. Viewing Fire and Detonation Effects

VR-Forces can display the following graphic effects when munitions are fired and detonated:

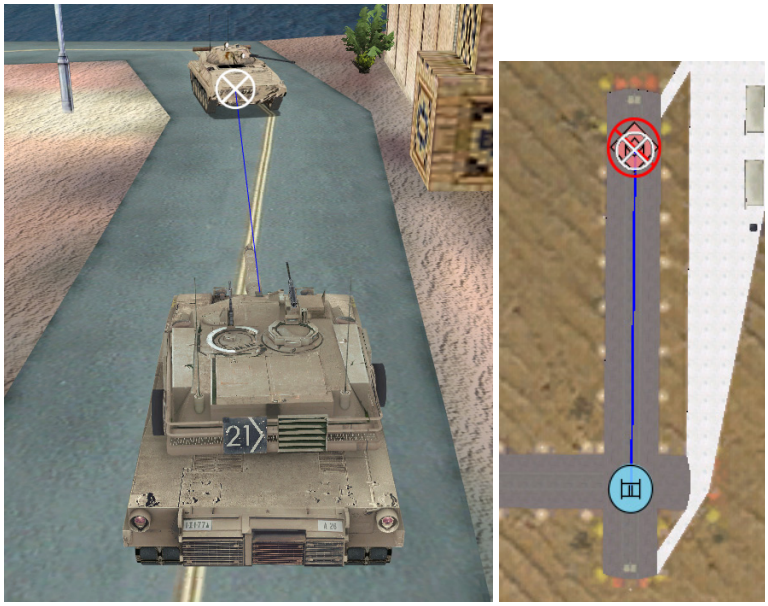
- Flaming effects.
- Smoke plumes.
- Detonations.
- Muzzle flashes.

You control display of these effects from the Observer menu or Observer Settings toolbar.

➤ To enable or disable display of a fire or detonation effect, choose **Observer** → *effect*, where *effect* is Flames, Smoke Plumes, Detonations, or Muzzle Flashes, or click the appropriate button on the Observer Settings toolbar.

8.8. Viewing Fire and Detonation Lines

If fire and detonation lines are enabled, when an entity fires a munition, a line is drawn between the entity and the location of the detonation. The line is colored for the force of the shooter (Figure 8-10).



3D 2D

Figure 8-10. Fire line and detonation

A detonation is identified by a circle with an X through it. The detonation color indicates the result of the detonation, as follows:

- Entity impact – white.
- Ground impact – gray.
- Airburst – black.
- Other – white.

➤ To enable or disable display of fire and detonation lines, choose **Settings** → **Fire / Detonate Lines** or click the Fire/Detonate Lines button (🚀) on the Display Settings Toolbar.

8.9. Viewing Aggregate Combat Graphics

When entities in aggregate-level scenarios engage in combat, they display two types of graphics, attack indicators, and attack animations. Attack indicators are triangular areas that show the direction an aggregate is attacking. They are colored with the force of the attacker. Attack animations show where an entity is being attacked. The footprint sector where the attack is taking place blinks from green to white.

The graphics persist until the attack is over.

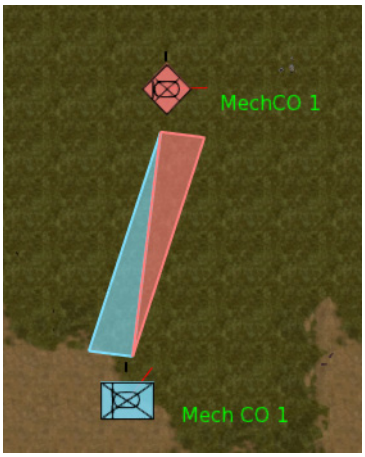


Figure 8-11. Attack indicators

To enable or disable aggregate combat graphics:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Aggregate Display Settings page (Figure 8-12).

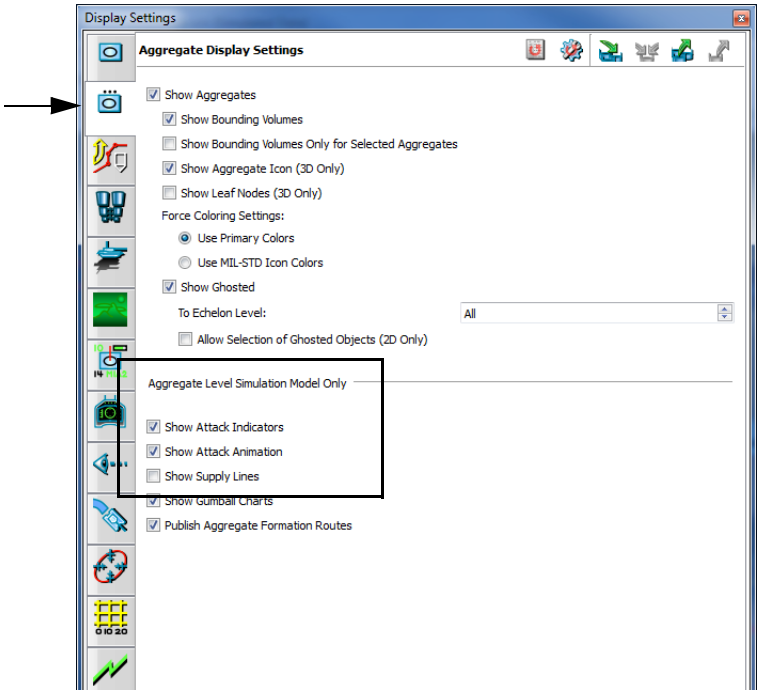


Figure 8-12. Aggregate Display Settings page

3. To enable or disable attack indicators, select or clear the Show Attack Indicators check box.
4. To enable or disable attack animations, select or clear the Show Attack Animations check box.

8.10. Displaying Laser Designators

You can display laser designation information if it is available. VR-Forces displays a green circle and a range (on top of the entity icon or model) (Figure 8-13).

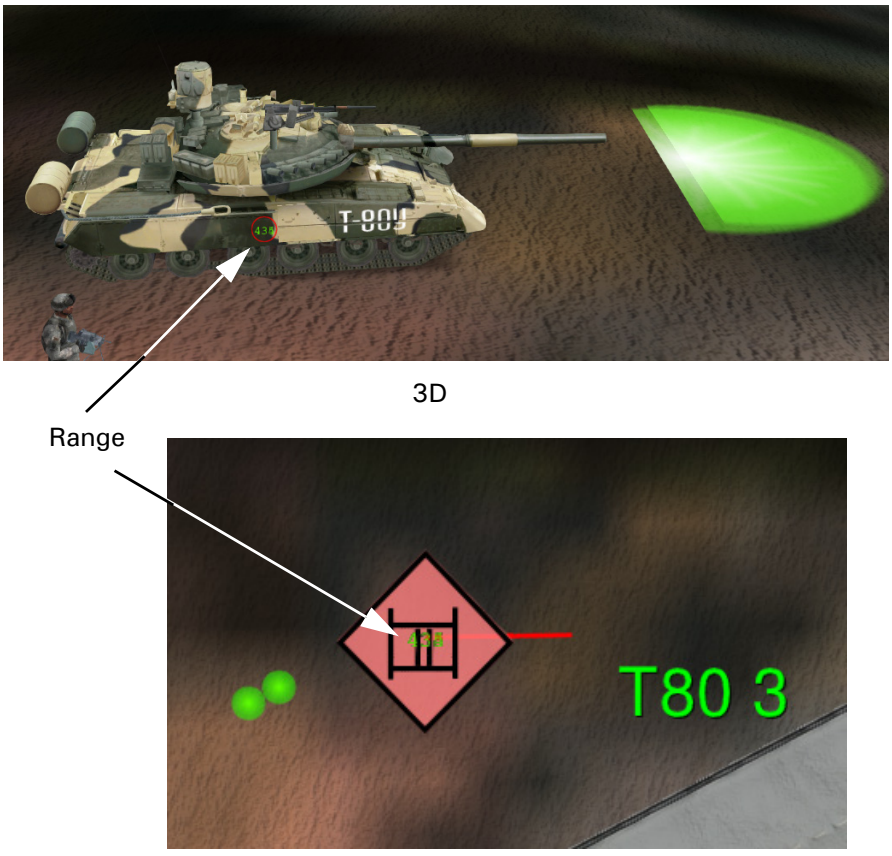


Figure 8-13. Laser designators

➤ To display laser designators, on the Display Settings Toolbar, click the Laser Designator button ( ).

8.11. Displaying Supply Lines

Entities in aggregate-level scenarios can provide supplies to and receive supplies from other entities. When an entity is supplying another entity you can display supply lines that show resupply is taking place.

To display supply lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Aggregate Display Settings page (Figure 8-12).
3. Select or clear Show Supply Lines.

8.12. Displaying Tactical Smoke

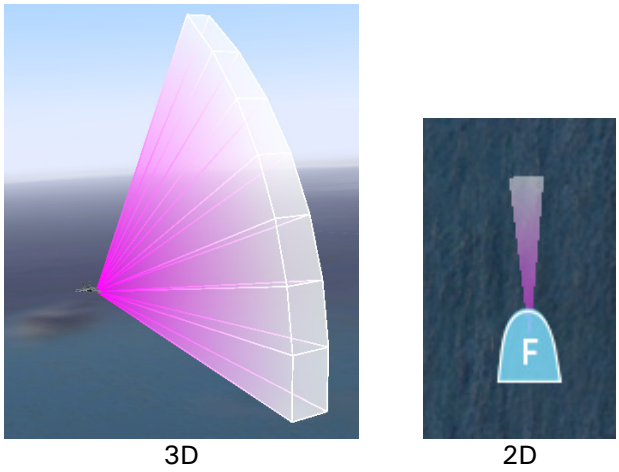
VR-Forces can display tactical smoke munitions fired by VR-Forces entities. You may want to disable tactical smoke if it obscures the observer's view. Also, if many entities are generating smoke, it could affect performance.

- Do not confuse tactical smoke and smoke plumes. Smoke plumes are a visual effect displayed when an entity is on fire. Tactical smoke visualizes a smoke munition that affects line-of-sight.
- If you disable display of tactical smoke, VR-Forces continues to simulate its existence in the back-end and it will continue to affect line-of-sight.

➤ To enable or disable display of tactical smoke, choose **Observer** → **Tactical Smoke**, or click the Tactical Smoke button ( ) on the Observer Settings toolbar.

8.13. Displaying Electromagnetic Emissions

VR-Forces can display electromagnetic emissions (emitter volumes) for entities that have emitter systems (Figure 8-14). You can configure the colors used to represent emitter frequencies and you can configure the size of the emitter volumes. (For details about configuring emitter volumes, please see Section 10.7, "Configuring Emitter Volumes," in *VR-Forces Configuration Guide*.)



3D 2D

Figure 8-14. Emitter volumes

Display of emitter volumes only affects entities that have emitter systems.

To enable or disable display of electromagnetic emissions:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 9-4).
3. Select or clear the Show Emitter Volumes check box.



You can also enable or disable emitter volumes by choosing **Settings** → **Emitter Volumes**, or clicking the Emitter Volumes button () on the Display Settings Toolbar.

8.14. Displaying Radio Communication Lines

VR-Forces can display squawk indicators when an entity sends a signal interaction. It can also display communication lines between entities when a VR-Forces entity (or an entity in an application built with the appropriate MÄK libraries) sends a radio message to another entity. You can configure a communication line's color and lifetime. Figure 8-15 displays squawk indicators and a line-of-sight communication line.



Figure 8-15. Radio communication line and squawk indicators

For communications over large distances, a line-of-sight communication line would travel through the earth. Therefore, you can also display communication lines drawn with the curvature of the earth (Figure 8-16).

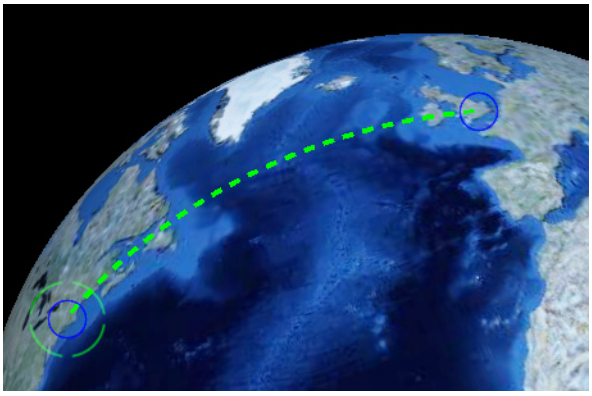


Figure 8-16. Curvature of the earth communication line

8.14.1. Enabling Radio Communications Graphics

You can globally enable or disable the ability to display radio communication lines. When they are enabled, you can enable and disable communication lines and squawk indicators separately.

To enable or disable radio communications graphics:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Radio Communications Settings page (Figure 8-17).

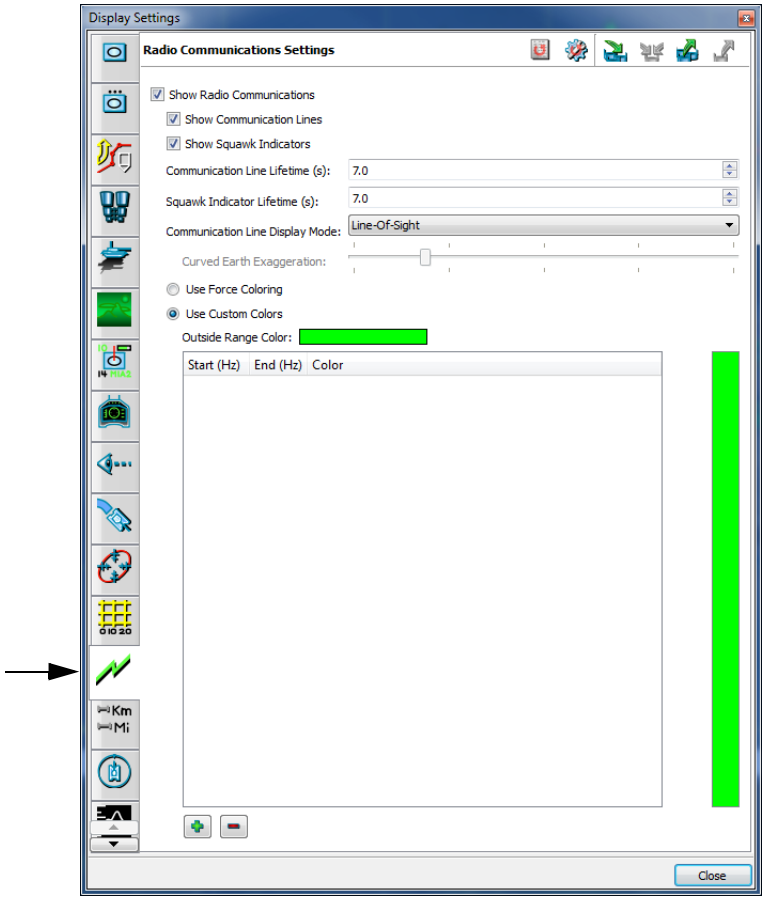


Figure 8-17. Radio Communications Settings page

3. Select or clear the Show Radio Communications check box.



You can also enable and disable communication lines by choosing **Settings** → **Radio Communication Display** or by clicking the Radio Communications button (⁄⁄) on the Display Settings Toolbar.

To enable or disable communication lines:

1. Enable radio communication lines.
2. Select or clear the Show Communication Lines check box.

To enable or disable squawk indicators:

1. Enable radio communication lines.
2. Select or clear the Show Squawk Indicators check box.

8.14.2. Configuring Radio Communication Lines Lifetime and Display Mode

You can configure how long a radio communication line is displayed. You can also configure whether it is displayed using line-of-sight or curvature of the earth.

To configure the lifetime and display mode for radio communication lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Radio Communications Settings page (Figure 8-17).
3. To specify how long a communication line or squawk indicator should be displayed, enter a value, in seconds, in the Communication Line Lifetime and Squawk Indicator Lifetime fields.
4. To specify the display mode, select on option in the Communication Line Display Mode list.
5. If you select Curvature of the Earth, adjust the Curved Earth Exaggeration slider to change the degree to which the curvature is exaggerated.

8.14.3. Configuring the Color of Radio Communication Lines

By default, radio communication lines are displayed is green lines. You can configure communication lines to use force coloring or to be colored based on the frequency of the message. When you color by frequency, you specify colors for the frequency ranges you are interested in and a color to use for all messages outside the specified frequencies.

To configure the color of radio communication lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Radio Communications Settings page (Figure 8-17).
3. To color radio communication lines using the force color, select the Use Force Coloring option.
4. To color radio communication lines by frequency:
   a. Select the Use Custom Colors option.
   b. Click the Add button (). A range is added to the frequency list with a default color (Figure 8-18).

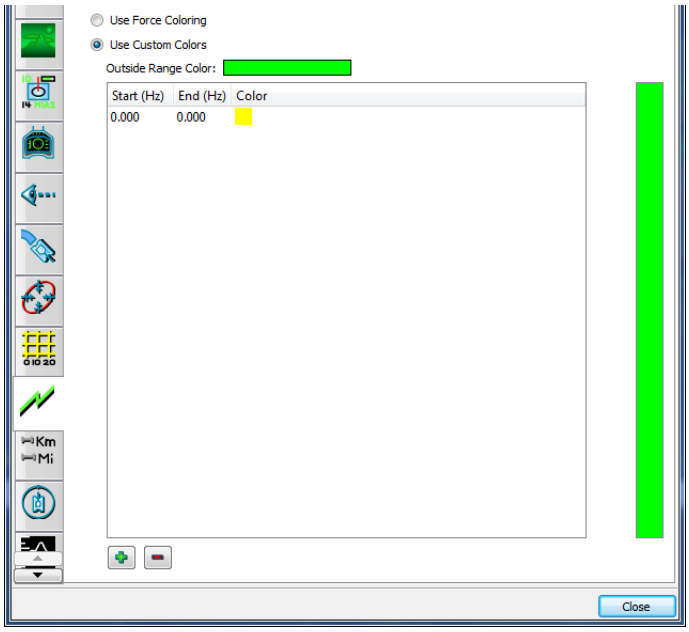


Figure 8-18. New frequency point

   c. Double-click the starting value of the range to make it editable (Figure 8-19).

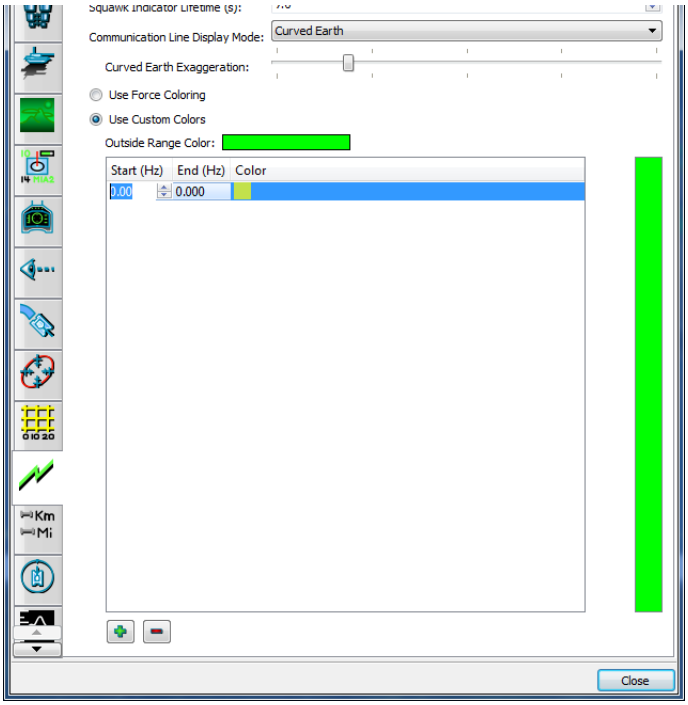


Figure 8-19. Editable frequency point

d. Type the starting frequency you want to use.

e. Edit the end frequency.

f. Double-click the color swatch and select a color for this range.

g. Optionally, add points for different frequency ranges.

h. Optionally, click the Outside Range Color color swatch and specify a color for all frequencies not specified in the frequency list.



If you configure multiple frequency ranges, you cannot add a range whose values are within those of an existing range.

8.15. Displaying Cockpit Displays (3D Only)

VR-Forces can display cockpit displays created with GL Studio. It includes cockpits for fixed-wing, rotary-wing, and ground vehicle entities. As with other entity effects, cockpits have schemas and model definitions that you map to entity types. For details about cockpit display models, please see Section 17.6, "Adding New Cockpit Display Models," in *VR-Forces Configuration Guide*.

VR-Forces supports 2D cockpits (Figure 8-20) and 3D cockpits (Figure 8-21). A 2D cockpit is fixed to the center of the screen and you cannot zoom in or out on the instrument panel. A 3D cockpit is aligned with the front of the vehicle. If you move the eyepoint, the instrument panel moves in or out of view as you would expect. You can zoom in and out on the instruments.



Figure 8-20. 2D cockpit



Figure 8-21. 3D cockpit

- Cockpits are displayed only when the observer is attached to an entity in Mimic mode.
- VR-Forces includes one 3D cockpit, for the F-16 fighter jet.
- Cockpits are not supported on Linux.

To display cockpits:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Cockpit Display Settings page (Figure 8-22).

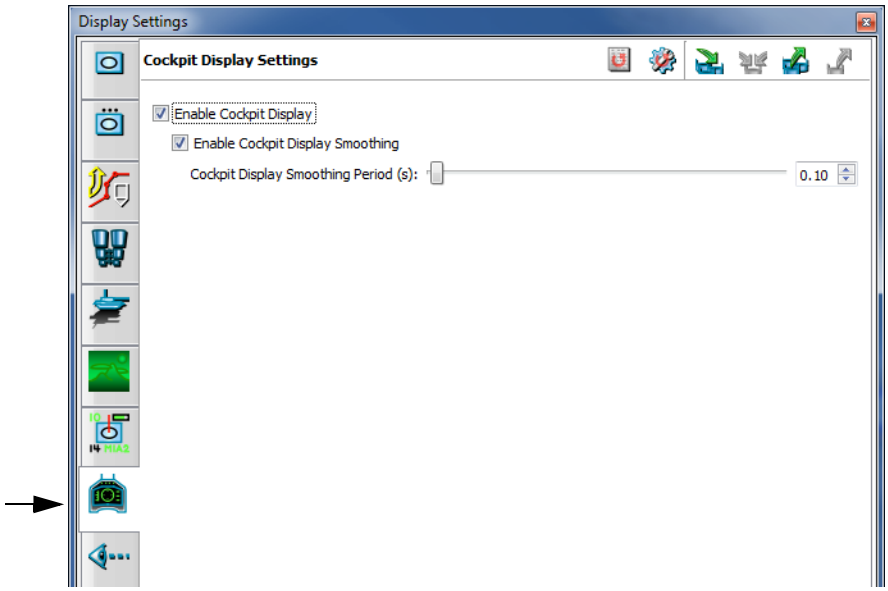


Figure 8-22. Cockpit display settings

3. To enable or disable cockpit displays, select or clear the Enable Cockpit Display check box.
4. Attach the observer to an entity in Mimic mode.

You can also enable and disable cockpit displays by choosing **Settings** → **Cockpit Displays** or by clicking the Cockpit Displays button ( ) on the Display Settings Toolbar.

8.16. Displaying Instrument Panels

Instrument panels are inset windows that display a vehicle instrument such as an altimeter or artificial horizon. Unlike a cockpit display, which is a fixed display that operates in Mimic mode, an instrument panel is an independent window (like an inset view) that you can place anywhere on the screen (Figure 8-23). It does not affect the observer mode. You can organize multiple instrument panels to create your own virtual cockpit.



Figure 8-23. Instrument panels

When you create an instrument panel, the title bar of the window lists the entity for which you created it.

- Instrument panels are GL Studio controls.
- Instrument panels are not supported on Linux.
- If the source of the instrument panel is removed from the exercise, for example when a Logger recording rewinds, the instrument panels are removed from the inset windows. To see them again you must open new instrument panels. Because the relationship between an instrument panel window and an entity is not persistent across multiple runs of an exercise, you cannot save an instrument panel layout for an entity. (You could save the layout of the windows to a display configuration, but they would not display an instrument panels when you loaded the configuration.)
- If you want to put multiple instrument panels in one window, like the six-pack panel provided with VR-Forces, you must use GL-Studio.

To open an instrument panel:

1. Select the entity for which you want to display an instrument panel.
2. On its right-click menu, choose **Create Instrument Panel**. A submenu is displayed.
3. Choose the instrument panel you want to open.

8.16.1. Configuring Entities to use Instrument Panels

VR-Forces includes a set of instrument panels for fixed-wing aircraft, although many of them may be appropriate for rotary-wing aircraft, and even for ground vehicles. (For example, you might want to show a compass for a ground vehicle to get its heading or an altimeter to show its height above sea level.) The availability of instrument panels to an entity is determined by the **instrumentPanelCategory** parameter in its model definition.

The instrument panels provided with VR-Forces have the **instrumentPanelCategory** set to FixedWing. All fixed-wing entities have this parameter set in their model definition. None of the other entities do. If you want other entities to display instrument panels provided with VR-Forces, you can add this parameter to their model definitions. (You could also copy an instrument panel model definition and change the category to some other value if you want to. Regardless, to use an instrument panel, an entity's model definition must include the **instrumentPanelCategory** parameter and set it to a valid value.)

If you create new instrument panels, you must add instrument panel model definitions for them and specify a value for the **instrumentPanelCategory** parameter. Then you can add the parameter to the appropriate entity model definitions.

8.17. Displaying Shadows (3D Only)

VR-Forces can display shadows for entities, props, and terrain features.



Figure 8-24. Shadows

You can enable and disable shadows and you can configure shadow quality.

Enabling shadows can affect performance.

To enable and disable shadows globally:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Shadow Settings page (Figure 8-25).

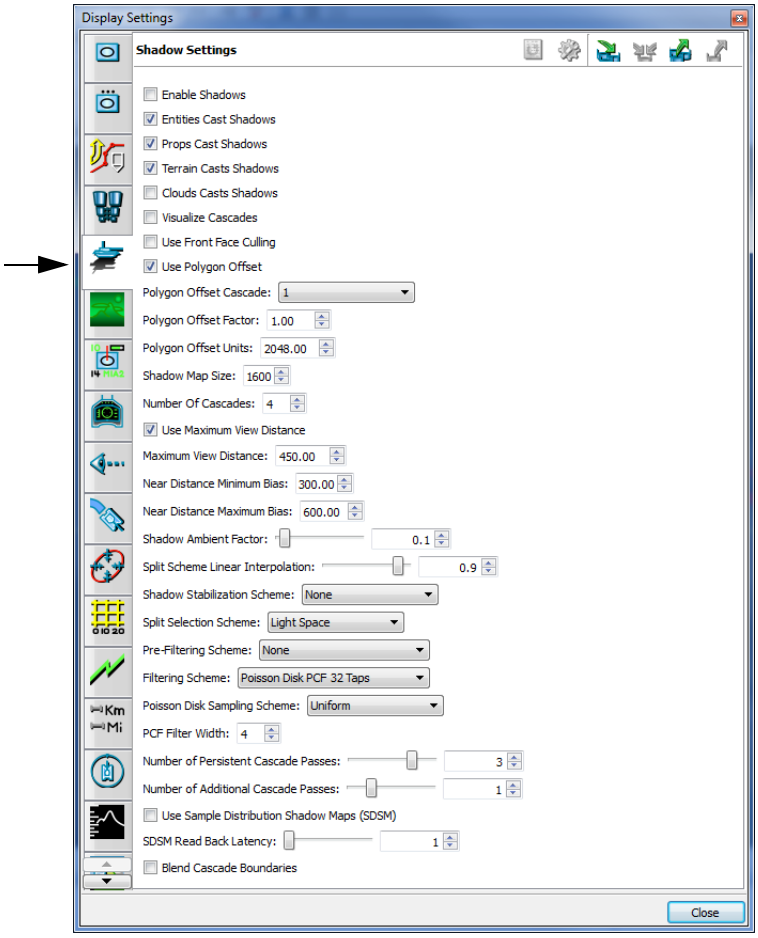


Figure 8-25. Display Settings, Shadow Settings page

3. Select or clear the Enable Shadows check box.



You can also enable and disable shadows by choosing **Settings** → **Shadows** or clicking the Shadows button () on the Display Settings Toolbar.

8.17.1. Configuring Shadows

You can configure which objects cast shadows and parameters that affect performance and the quality of shadows. Shadow parameters include the following:

- Cascades. A scene is split into sections, with those closest to the camera showing the highest resolution shadows. You can configure the number of splits and how the quality of resolution changes as you move away from the camera.
- Polygon offset. These settings shift shadows to minimize Z-fighting.
- Shadow ambient factor. Hides Z-fighting and other visual artifacts.
- Shadow stability. Affects resolution when the camera moves.
- Filtering. These options affect visual quality.
- Sample Distribution Shadow Maps (SDSM). SDSM increases rendering quality at the expense of performance.

To configure how shadows are rendered:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Shadow Settings page (Figure 8-25).
3. To enable or disable shadows for object types, select or clear the Entities Cast Shadows, Props Cast Shadows, or Terrain Casts Shadows check boxes.
4. Change other parameters as desired.

8.18. Using the Depth of Field Effect

Depth of field controls the area in the scene that is in focus. Depth of field is calculated based on a focal depth and a focal range. The focal depth is the distance from the observer that is the center (focal point) of the area in focus. The focal range is the distance from the focal point that is in focus.

In Figure 8-26, the focal depth is about 10 meters, fairly close to the observer, and the focal range is also small, so that the scene is out of focus just past the car (note that the first large tree is out of focus). In Figure 8-27, the focal depth is the same, but the range is much greater, so the scene is in focus to the back of the parking lot.

In Figure 8-28 the focal depth is centered at the second large tree. The focal range is, again, fairly narrow, so that the person and car in the foreground are out of focus.

VR-Forces lets you set the focal depth automatically or manually. In either case, you can adjust the focal range.



Figure 8-26. Depth of field (close)



Figure 8-27. Depth of field (far)



Figure 8-28. Depth of field, middle distance

To enable depth of field:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 8-6).
3. Select the observer mode for which you want to enable depth of field.
4. Select or clear the Depth of Field check box.



You can also enable or disable depth of field by choosing **Observer** → **Depth of Field** or clicking the Depth of Field button ( ) on the Observer Settings Toolbar.

8.18.1. Configuring Depth of Field

You can enable and disable automatic depth of field and you can adjust the focal depth and focal range.

To configure depth of field:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Render Settings page (Figure 8-29).

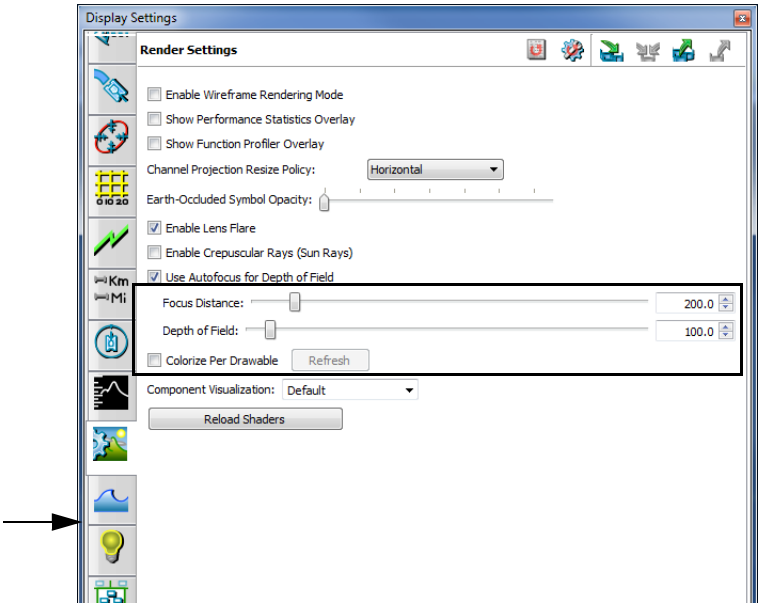


Figure 8-29. Depth of Field settings

3. To use autofocus for focal depth, select the Use Autofocus for Depth of Field check box.
4. To set the focal depth manually, clear the Use Autofocus for Depth of Field check box and adjust the Focal Distance slider or specify a value in the box.
5. To adjust the focal range, adjust the Depth of Field slider or specify a value in the box.

8.19. Displaying Wakes and Spray Effects

When dynamic ocean is enabled, surface entities display wakes and spray effects. However, these effects affect performance. You can disable them if you need to emphasize performance over visual effects.

To enable or disable wakes and spray effects:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 8-30).

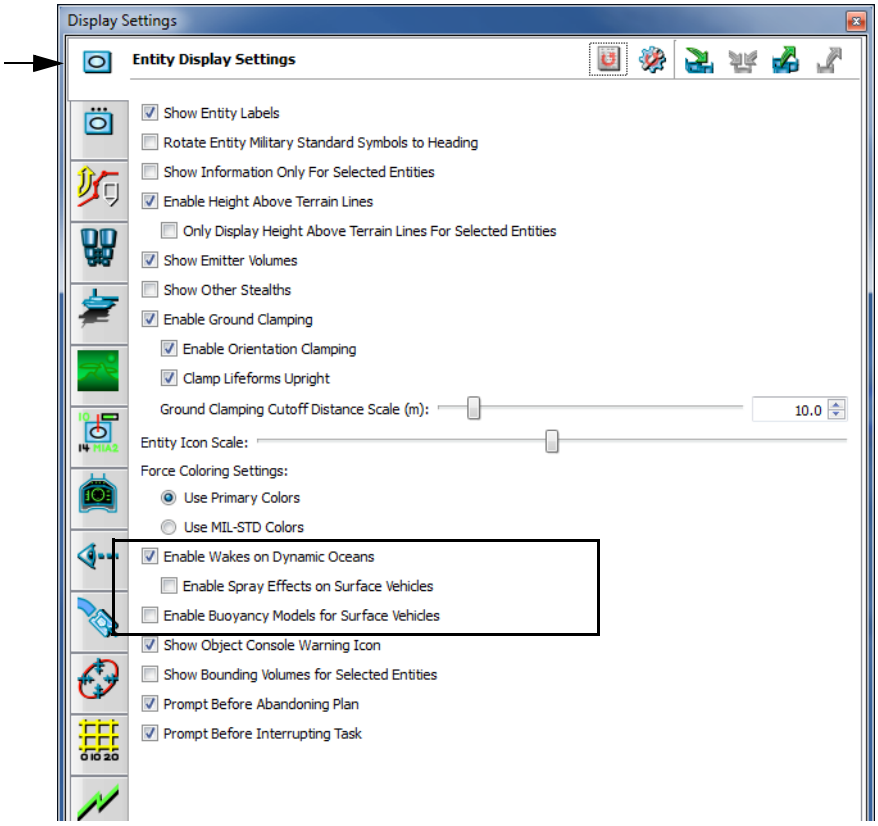


Figure 8-30. Entity Display Settings page

3. To enable or disable wakes, select or clear the Enable Wakes on Dynamic Oceans check box, or toggle the Wake button ( ) on the Display Settings Toolbar.
4. If wakes are enabled, select Enable Spray Effects on Surface Vehicles to enable spray effects. Clear the check box to disable them. Disabling wakes also disables spray effects.

8.20. Enabling Buoyancy for Surface Entities

When dynamic ocean is enabled, surface entities can bob up and down with wave action. The buoyancy models affect performance. You can disable buoyancy if you need to emphasize performance over visual effects. You can also specify a distance beyond which buoyancy models are not used. The farther the distance, the more that buoyancy affects performance.

To enable or disable the buoyancy models:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 8-30).
3. Select or clear the Enable Buoyancy Models for Surface Vehicles check box.
4. To change the buoyancy LOD distance, adjust the Default LOD Distance slider or change the value in the box.



9. Displaying Aggregate Entities

Many aspects of managing aggregate entities are the same as for individual entities. This chapter focuses on features that are unique to aggregates.

Selecting Aggregates 9-2
Expanding and Collapsing Aggregates 9-2
Expanding and Collapsing Aggregates in the Echelon View 9-4
Expanding and Collapsing the Echelon View by Level of Aggregation 9-4
Displaying Ghosted Icons 9-6
Displaying Aggregate Icons and Bounding Volumes 9-9
Specifying the Color Scheme for Aggregate Bounding Volumes 9-11

9.1. Selecting Aggregates

You can select aggregates in the same ways that you can select entities. (For details about selecting entities, please see "Selecting Entities, Tactical Graphics, and Props," on page 6-4.) However, to select an aggregate, you must select the aggregate icon, not individual members of the aggregate. If a disaggregated aggregate is expanded, the aggregate icon might not be visible. To select the aggregate, you can collapse it, or you can select the aggregate level entity in the Echelon View.

9.2. Expanding and Collapsing Aggregates

You can display disaggregated aggregates in expanded or collapsed mode. When an aggregate is collapsed, it is displayed as one icon. When an aggregate is expanded, all the members of the aggregate are visible as individual entities and the aggregate icon is hidden. Multi-level aggregates can expand and collapse at each level of aggregation. For example a company could be collapsed to the company level (one icon), the platoon level (an icon for each platoon), or fully expanded.



Aggregate-level scenarios do not support aggregation and disaggregation of aggregates. Aggregates are always disaggregated. Therefore, they can always be expanded and collapsed.

When you Expand or Collapse an aggregate, you can expand or collapse one level of aggregation or all levels. When you Expand or Expand All, expansion is downward from the current aggregate and does not affect any sibling aggregates. When you Collapse, contraction is upward in the hierarchy. Collapse All collapses all members of the aggregate to the topmost level.

Figure 9-1 illustrates these relationships. Assume a company with three platoons, each of which has four vehicles.

- If you Expand the top level aggregate, it expands to the platoon level. (1)
- If you select Plt C and choose Expand or Expand All, it expands just Plt C. (2)
- If you select the top level aggregate and expand all, it expands to the leaf level – 12 individual vehicles. (3)
- If the aggregate is fully expanded and you select one member of Plt C and choose Collapse, just Plt C collapses. (4)
- If you select any member of an expanded aggregate and select Collapse All, the entire aggregate collapses. (5)

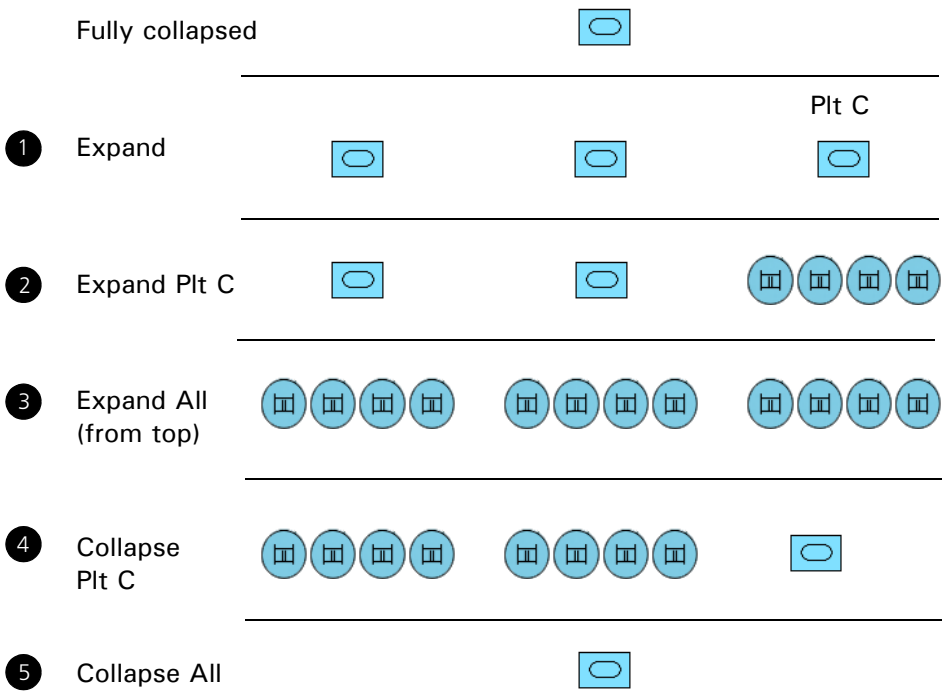


Figure 9-1. Aggregate expand and collapse



An aggregate's icon does not indicate whether it is aggregated or disaggregated. However you can easily infer its state from the Entities menu or the Echelon View. The aggregate commands on the Entities menu are context sensitive. If you select an aggregate and the Disaggregate command is available, then the aggregate is in the aggregated state. Similarly, if you view an aggregate in the Echelon View, if it is aggregated, you cannot expand it in the tree. If it is disaggregated, you can expand it to see its subordinates.

- To expand an aggregate, right-click the aggregate icon or its entry in the Objects List Panel and choose **Expand** on the popup menu.
- To expand an aggregate and all its subordinate aggregates, right-click the aggregate icon or its entry in the Objects List Panel and choose **Expand All** on the popup menu.
- To collapse an aggregate one level, right-click any member of the aggregate and choose **Collapse** on the popup menu.
- To collapse an aggregate to its root level, right-click any member of the aggregate and choose **Collapse All** on the popup menu.

9.2.1. Expanding and Collapsing Aggregates in the Echelon View

You can expand and collapse aggregates in the Echelon View on the Objects List Panel.

➤ To expand an aggregate in the Echelon View, click the plus sign next to its icon.

➤ To collapse an aggregate in the Echelon View, click the minus sign next to its icon.

9.2.2. Expanding and Collapsing the Echelon View by Level of Aggregation

In addition to expanding and collapsing the order of battle on a node-by-node basis, as in typical hierarchical view windows, the Echelon View enables you to view entities by the level of aggregation.

Individual entities are directly subordinate to the force they belong to. If an entity becomes part of an aggregate, the aggregate is subordinate to the force and the individual entity is one level further away. Seen another way, you can say that the force is the parent and all other entities in that force are children, grandchildren, and so on.

The Echelon View displays the entities in columns that are equivalent to their relationship to the force – the first column is children, the second column is grandchildren, and so on.

At the top of each column in the Echelon View is a plus (+) or minus (-) button that works like the buttons within the hierarchical display. A plus indicates that at least one entity in that column is an aggregate entity and that none of the aggregates in the column are expanded. A minus indicates that at least one aggregate at that level is expanded. Clicking a minus sign collapses all aggregate entities in the column. Clicking a plus sign expands all aggregates in the column.

➤ To change the hierarchical level view in the Echelon View, click the level controls in the horizontal band across the top of the window.

Figure 9-2 illustrates an Echelon View with several levels of aggregates, some of which are collapsed and some of which are expanded.

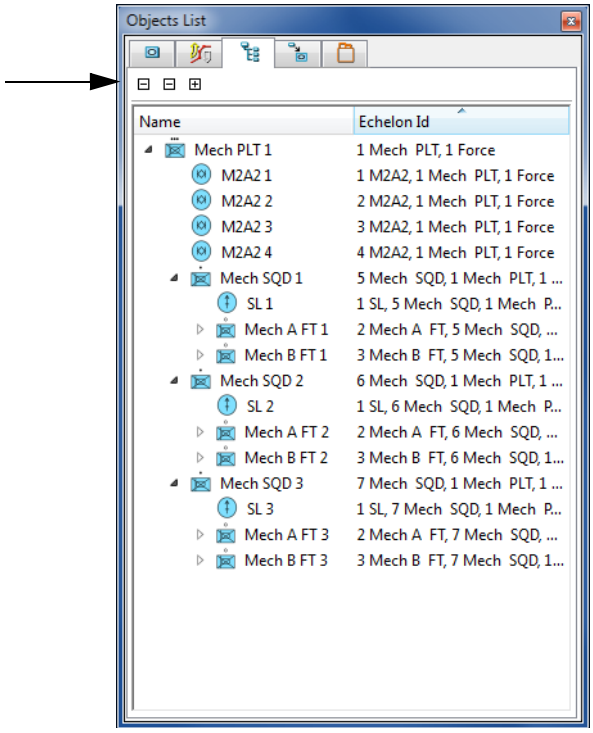


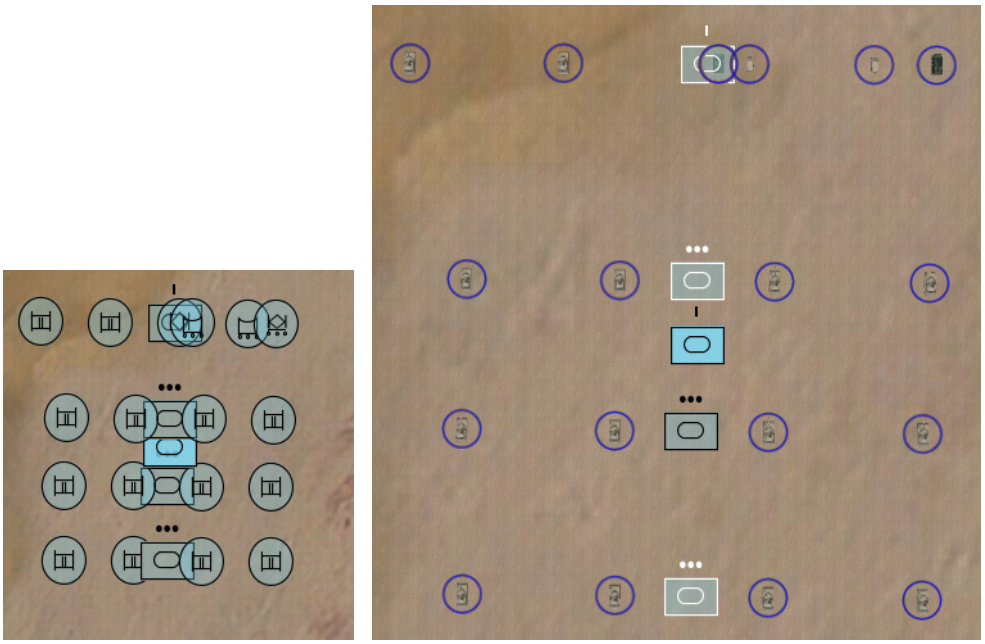
Figure 9-2. Echelon View

9.3. Displaying Ghosted Icons

If you want to select aggregates, you need to collapse them so that you can select the aggregate icon. However, when you do this, you can no longer see where individual subordinate entities are located and if subordinates are destroyed, you cannot see which ones are destroyed and which are functional. VR-Forces provides the following options for how aggregate icons are displayed so that you can work with them in a way that best meets your visualization and entity management needs:

- Display ghosted icons for hidden entities.
- Allow ghosted icons to be selected.
- Display all aggregate members.
- Manually expand and collapse the display.
- Display leaf-level aggregate members.

In the Plan View observer mode, when you display ghosted aggregates, the aggregate icon is displayed normally, and the subordinate members are displayed using semi-transparent icons. In 3D observer modes, aggregates that are subordinate to the top level aggregate are displayed using semi-transparent icons. Leaf-level entities are displayed using the appropriate model for the observer mode (Figure 9-3).



2D 3D

Figure 9-3. Ghosted icons

To configure the display of aggregate icons:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Aggregate Display Settings page (Figure 9-4).

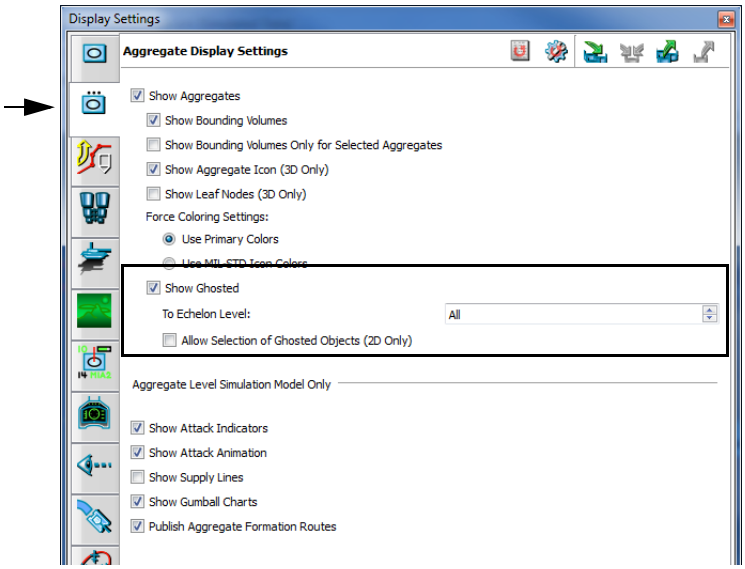


Figure 9-4. Aggregate Display Settings page

3. To display aggregate members based on the current level of aggregation only, clear Show Ghosted.
4. To display ghosted icons:
   a. Select Show Ghosted.
   b. In the To Echelon Level box, select the level of aggregation to show ghosted icons for. To show all levels without needing to consider how many levels there are, select All. Figure 9-5 illustrates the effect of changing the echelon level for displaying ghosted icons for the VR-Forces aggregate hierarchy shown.
   c. To allow selection of ghosted icons (2D view only), select Allow Selection of Ghosted Icons. (In 3D views, you can always select entities at the leaf level.)

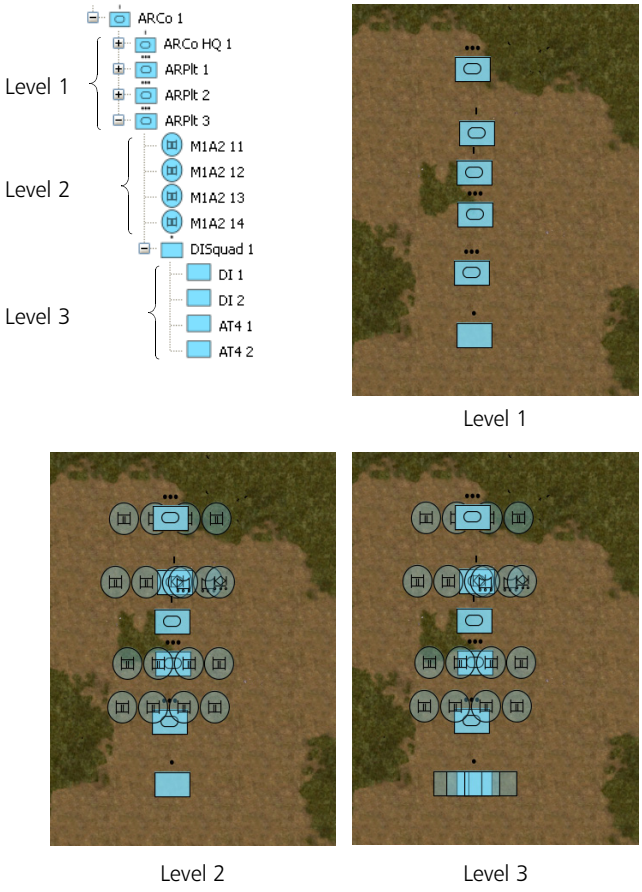


Figure 9-5. Displaying ghosted icons by echelon level

9.4. Displaying Aggregate Icons and Bounding Volumes

VR-Forces lets you configure when to display aggregate icons and bounding volumes. Bounding volumes are translucent polygons that show the extents of the aggregate. This is useful when the individual entity icons or models are not displayed. You can configure icons and bounding volumes as follows:

- Enable and disable the display of aggregates. When disabled, only the 2D icons or 3D models for leaf level aggregate members are displayed.
- Enable and disable the display of bounding volumes. Bounding volumes are displayed only for collapsed aggregates. You can restrict the display of bounding volumes to the selected aggregates.
- In 3D observer modes, enable or disable the display of aggregate icons.

Figure 9-6 shows an aggregate bounding volume in Stealth observer mode, with leaf nodes displayed.

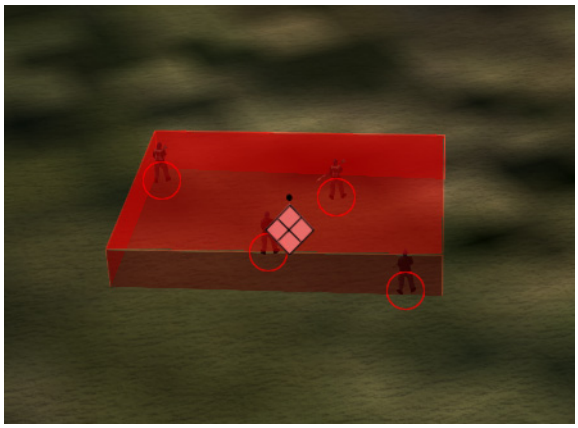


Figure 9-6. 3D bounding volume

Figure 9-7 shows an aggregate bounding volume in Plan View observer mode, with ghosting enabled.

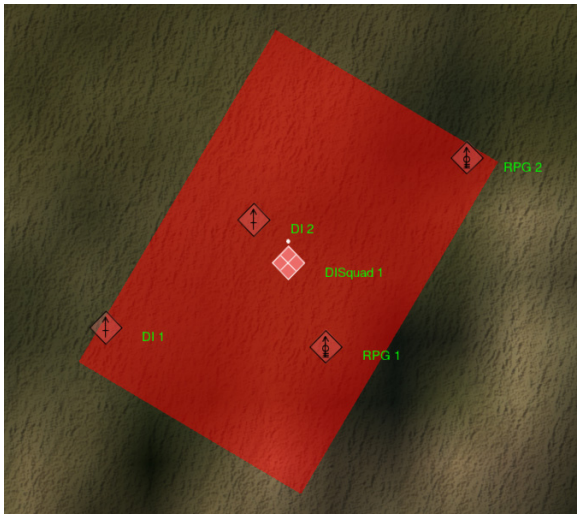


Figure 9-7. 2D bounding volume

You can display aggregate icons without showing bounding volumes and vice versa.

To configure the display of aggregate icons and bounding volumes:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Aggregate Display Settings page (Figure 9-4).
3. To enable the display of aggregate icons, bounding volumes, or both, select Show Aggregates.
4. To display bounding volumes for collapsed aggregates, select Show Bounding Volumes.
5. To restrict the display of bounding volumes to the selected aggregates, select Show Bounding Volumes Only for Selected Aggregates.
6. To display aggregate icons in 3D observer modes, select Show Aggregate Icon (3D Only).



You can also enable and disable aggregates by choosing **Settings** → **Show Aggregates** or by selecting the Show Aggregates button ( ) on the Display Settings Toolbar.

9.4.1. Specifying the Color Scheme for Aggregate Bounding Volumes

Aggregate bounding volumes can be displayed using primary colors or MIL-STD colors.

To specify the color scheme to use for aggregate bounding volumes:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Aggregate Display Settings page (Figure 9-4).
3. Select the Force Coloring Settings option that you want (Use Primary Colors or Use MIL-STD Icon Colors).



10. The Observer and Observer Modes

This chapter explains how to create, edit, and change observers and observer modes. For an introduction to observers and observer modes, please see "The Observer," on page 4-2. For details about moving the observer, please see Chapter 11, *Moving the Observer*. For details about attaching the observer to entities and props, please see Chapter 12, *Attaching the Observer to Objects*.

Changing the Observer Mode 10-2
Creating and Editing Observer Modes 10-2
Creating an Observer Mode 10-3
Deleting an Observer Mode 10-4
Editing an Observer Mode 10-4
Renaming an Observer Mode 10-4
Adding an Observer 10-5
Adding an Observer on the Observer Settings Page 10-5
Adding an Observer in the Display Engine Configuration Editor 10-6
Removing an Observer 10-8
Communicating Information About the Observer 10-8
Making the Observer Visible to Other Applications 10-8
Viewing and Attaching to Other Observers 10-9
Displaying Models for Remote Observers 10-9

10.1. Changing the Observer Mode

When you change the observer mode, terrain may be displayed differently and the model sets used to display entities and objects may change. For descriptions of the observer modes, please see "Observer Modes," on page 4-3.

➤ To change the observer mode, choose **Observer** → **Set Observer Mode** → *mode*, or on the Observer Settings Toolbar (Figure 10-1), choose an option from the Observer Mode list.



Figure 10-1. Observer Settings Toolbar

10.2. Creating and Editing Observer Modes

VR-Forces comes with 3D and 2D observer modes. They are configured with the display settings, key mappings, and model sets that users most commonly want to use. However, if they do not meet your needs, you can easily edit them or create new observer modes.

10.2.1. Creating an Observer Mode

To create an observer mode:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 10-2).

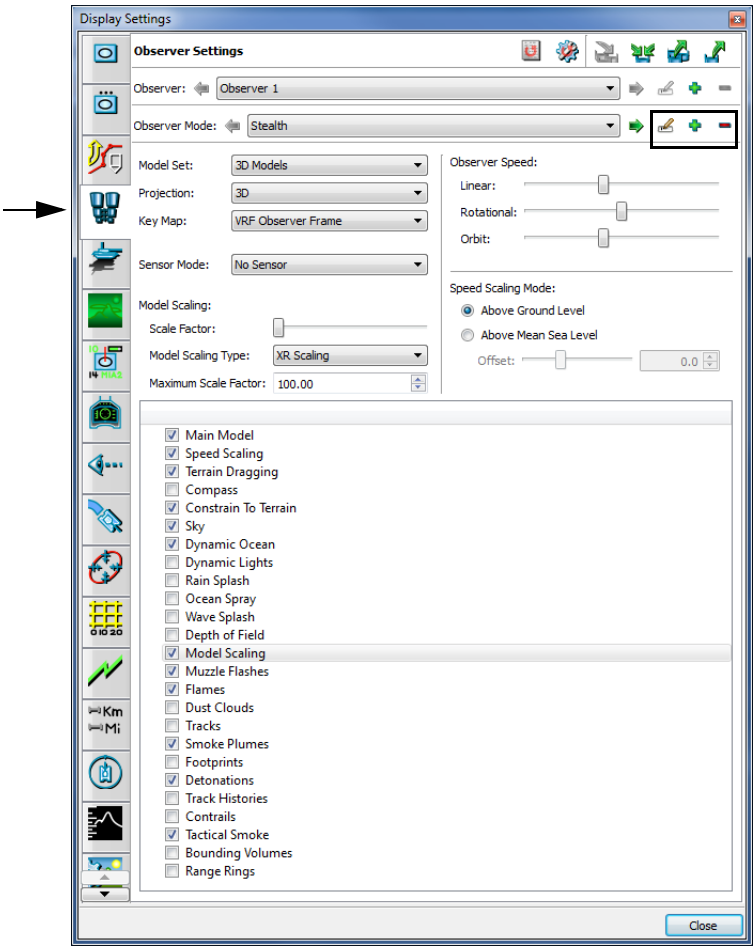


Figure 10-2. Observer Settings page

3. Click the Add button (+) to the right of the Observer Mode list. The Create New Observer Mode dialog box opens (Figure 10-3).

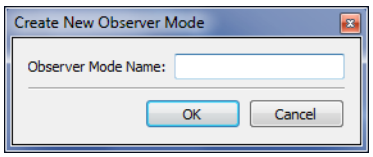


Figure 10-3. Create New Observer Mode dialog box

4. Type a name for the new observer mode.
5. Click OK.
6. In the Model Set list, select a model set.
7. Select a projection (2D or 3D).
8. Select the key map you want to use.
9. Enable or disable the mode-specific settings. The list of settings varies depending on the projection and model set.

10.2.2. Deleting an Observer Mode

To delete an observer mode:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 10-2).
3. In the Observer Mode list, select the observer mode that you want to delete.
4. Click the Delete button ( ).

10.2.3. Editing an Observer Mode

To edit an observer mode:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 10-2).
3. In the Observer Mode list, select the observer mode that you want to edit.
4. Edit the attributes of the observer mode as desired.

10.2.4. Renaming an Observer Mode

To rename an observer mode:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 10-2).
3. In the Observer Mode list, select the observer mode that you want to rename.
4. Click the Rename button ( ). The Rename Observer Mode dialog box opens.
5. Type the new name.
6. Click OK.

10.3. Adding an Observer

You can add and remove observers. Having multiple observers lets you assign different observers to different windows or channels. Changing the position and orientation of the observer in one channel does not affect the view of a different observer in a different channel.



If you have multiple observers, the observer listed in the Observer Settings Toolbar is the observer assigned to the window that has focus. If you open the Display Configuration Panel and select a different observer in the Channel Attributes list, the data in relevant panels will change to reflect the values for the selected observer. However, as soon as you return focus to a window, the observer selection will change to match that of the window.

You can add observers in the Display Engine Configuration Editor Panel or on the Observer Settings page.

If you want to save your observers, save the display engine configuration in which you created them. (Saving an observer just saves its name. It does not save any information about a particular observer mode or location that you may have used for that observer. To save location information, save an observer view.)

10.3.1. Adding an Observer on the Observer Settings Page

To add an observer on the Observer Settings page:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 10-2).
3. Click the Add button () to the right of the Observer list. The Create New Observer dialog box opens.
4. Type a name for the new observer.
5. Click OK. The new observer is added to the list.
6. To save the observer, save the display configuration. (Saving the observer settings does not save observers.)

10.3.2. Adding an Observer in the Display Engine Configuration Editor

To add an observer on the Display Engine Configuration Editor Panel:

1. Choose **View** → **Display Engine Configuration Editor Panel**. The Display Engine Configuration Editor Panel opens (Figure 10-4).

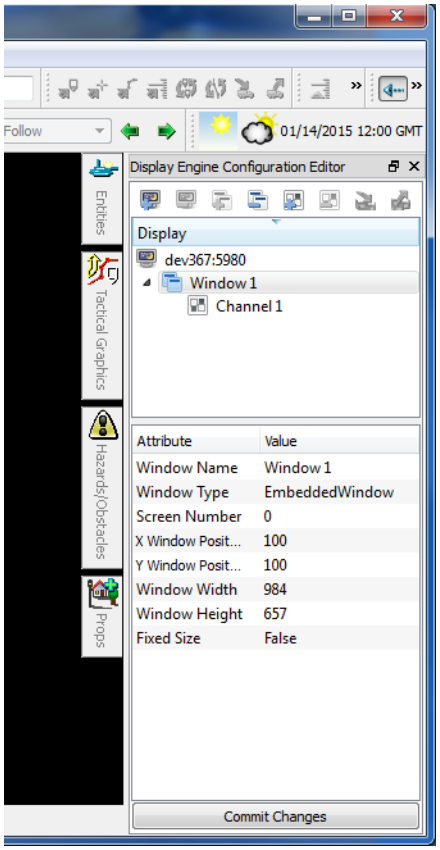


Figure 10-4. Display Engine Configuration Editor Panel

2. Select the channel on which you want to add an observer.
3. Click the Value field for the Observer Name attribute to make it editable.
4. In the list, select New (Figure 10-5).

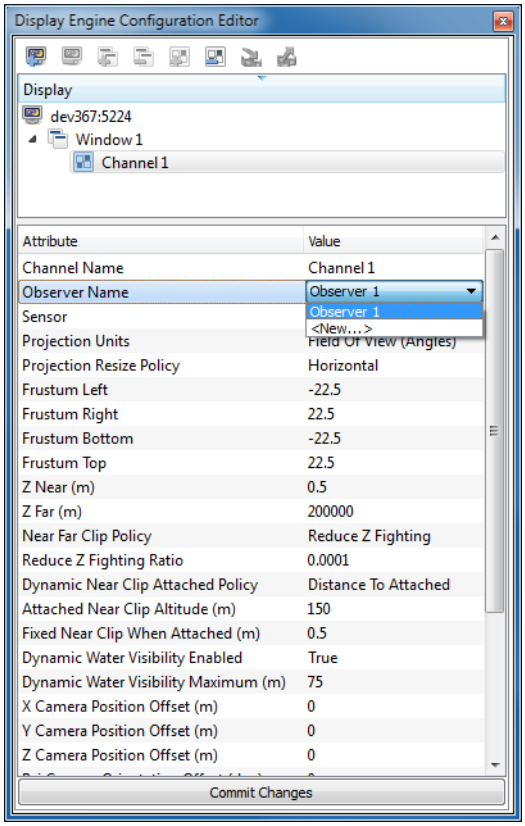


Figure 10-5. Observer Name attribute

5. Click Commit Changes. The Add Observer dialog box opens.
6. Type a name for the new observer.
7. Click OK. The new observer becomes the selected observer on the Observer Settings Toolbar, and is listed in the Observer Name attribute. (You may need to refresh the channel attributes before the observer name is listed.)
8. To save the observer, save the display configuration.

10.4. Removing an Observer

When you close VR-Forces, all observers get removed. They are not saved unless you save a display configuration. You can also remove observers during a session.

To remove an observer:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 10-2).
3. In the Observer list, select the observer that you want to remove.
4. Click the Delete button (−). The observer is removed.
5. If the observer was saved as part of a display configuration, save the updated display configuration.

10.5. Communicating Information About the Observer

VR-Forces can transmit state updates describing the observer's position and orientation, and can respond to view control messages sent by other applications.

10.5.1. Making the Observer Visible to Other Applications

By default, VR-Forces sends state update messages to make its state information known to other applications. This makes it possible for a plan view display to represent VR-Forces as an icon, or for one VR-Forces application to synchronize its observer movement with that of another VR-Forces application.

➤ To enable or disable broadcasting of observer information, choose **Settings** → **Publish Observers** (or **Stop Publishing Observers**).

Making the Observer Visible to Just One Application (DIS Only)

In DIS you can send observer information to just one other application, rather than to everyone in the exercise. To do this, on the Simulation Connections Configuration dialog box, instead of using a broadcast address, set the destination address to the address of a specific computer.

10.5.2. Viewing and Attaching to Other Observers

If an exercise includes stealth applications, and they are sending state updates, VR-Forces can detect them. A remote observer entity is listed in the object hierarchy, so you can attach the local observer to it just as you would attach to any other entity.

Observer icons are displayed in the 2D view, but they are not normally visible in the 3D view, even when they appear in the entity list. If you want to display visual representations of remote observers, map a visual model to the observer's entity enumeration.

For related information, please see "Displaying Models for Remote Observers," on page 10-9.

10.6. Displaying Models for Remote Observers

VR-Forces broadcasts the location and orientation of the observer. By default, it does not display a model for the observer. However, if there are other applications in an exercise that are broadcasting a stealth PDU (such as another VR-Forces), you might want to represent them with a model.

To display a model for remote observers:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Entity Display Settings page (Figure 10-6).

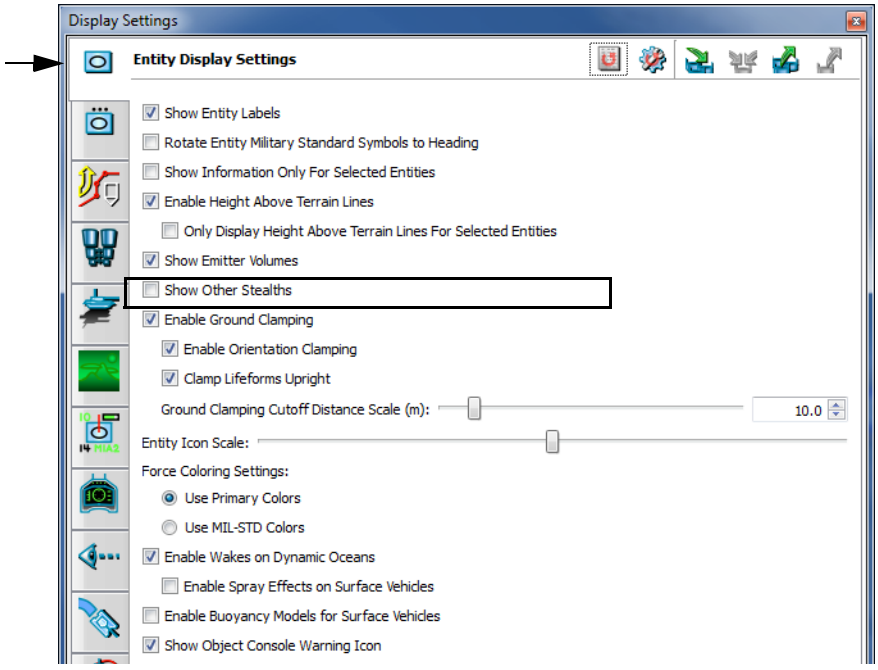


Figure 10-6. Display Settings, Entity Display Settings page

3. Select the Show Other Stealths check box.



11. Moving the Observer

This chapter explains how to move the observer (eyepoint) in the VR-Forces window.

Moving the Observer 11-3
Using the Compass to View the Observer's Heading 11-4
Editing the Compass's Attributes 11-5
Observer Movement Functions and Keyboard Mappings 11-6
3D Observer Movement Coordinate Systems 11-6
2D Observer Coordinate System 11-7
The Attached Context and Observer Movement (3D Only) 11-7
Moving the Observer from the Keyboard in the 3D View 11-9
Navigation Options 11-11
Orbiting 11-12
Moving the Camera 11-12
Moving the Observer Using the Mouse (3D View) 11-13
Dragging the Terrain 11-13
Changing the Observer's Orientation with the Mouse 11-14
Dragging the View 11-14
Orbiting the Terrain with the Mouse 11-14
Teleporting the Observer 11-15
Moving the Observer in the 2D View 11-15
Moving the Observer from the Keyboard in the 2D View 11-15
Moving the Observer Using the Mouse in the 2D Projection 11-16
Zooming the Observer 11-16
Zooming the Observer in the 3D Projection 11-17
Zooming the Observer in the 2D Projection 11-18
Setting the Scale Factor for Paged LODs 11-18

Zooming to Extents 11-18
Centering the Observer on an Entity 11-19
Forcing Orientation North 11-20
Enabling and Disabling View Constraints (3D Only) 11-21
Controlling Navigation Speed 11-21
Enabling or Disabling Speed Scaling 11-22
Changing the Observer's Movement Speed 11-23
Saving and Recalling Views 11-23
Recalling a View 11-25
Deleting Views 11-25
Specifying the Startup View 11-25
Saving Views to a File 11-26
Loading Views 11-26
Controlling the Observer from Other Applications 11-27
Enabling the Processing of View Control Messages 11-27

11.1. Moving the Observer

When you view a simulation, you usually want to move the observer (eyepoint) to different locations to view the terrain and scene elements being simulated. In VR-Forces, you can move the observer using the keyboard and the mouse. There are two basic contexts in which you can move the observer:

- You can move it independently of any scene element. This is called absolute, or free-fly mode.
- You can attach the observer to an entity, in which case it moves with the entity. (You can also attach the observer to a prop. For details about attaching the observer to an entity or prop, please see Chapter 12, *Attaching the Observer to Objects*.)

Absolute mode and the various attachment types are collectively called attach modes.

Regardless of which attach mode you are using, you can change the position and orientation of the observer. In the 3D view, position is defined by X, Y, and Z coordinates that identify its location in the 3D world. Orientation describes the observer's yaw (heading), pitch, and roll (Figure 11-1).

In the 2D view, the Z coordinate is fixed. You can change the X and Y coordinates. You can change the observer's yaw, but pitch and roll are fixed.

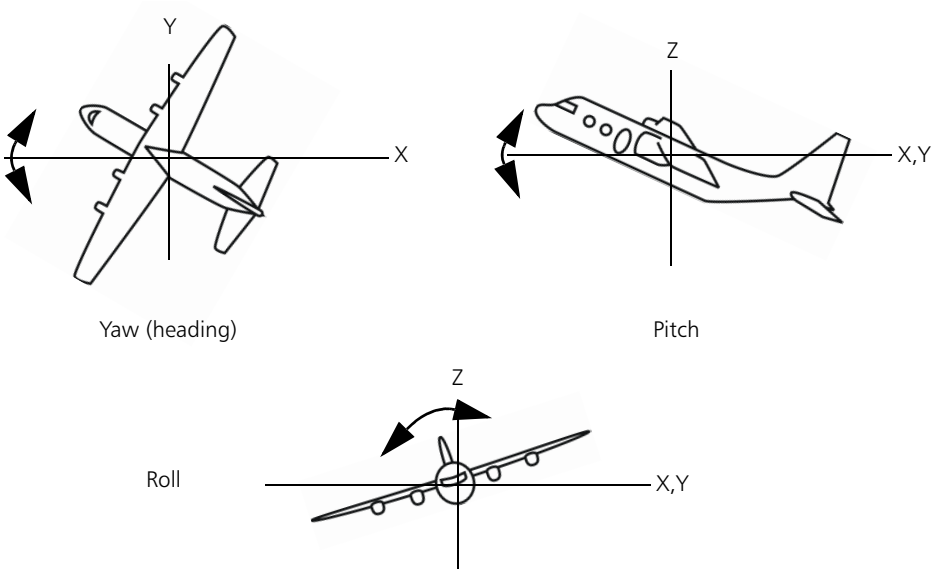


Figure 11-1. Orientation

The Observer Information Panel (Figure 11-4) displays the location and orientation of the observer in whichever coordinate system you select.

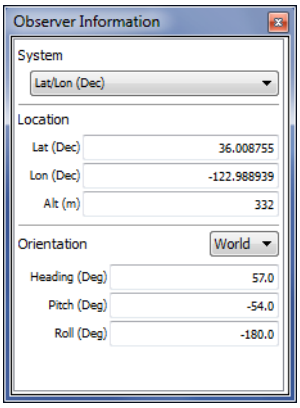


Figure 11-2. Observer Information Panel

11.1.1. Using the Compass to View the Observer's Heading

VR-Forces can display a compass that always shows the heading of the observer. The compass is an observer mode setting. In Figure 11-3, the observer's heading is approximately Northwest.

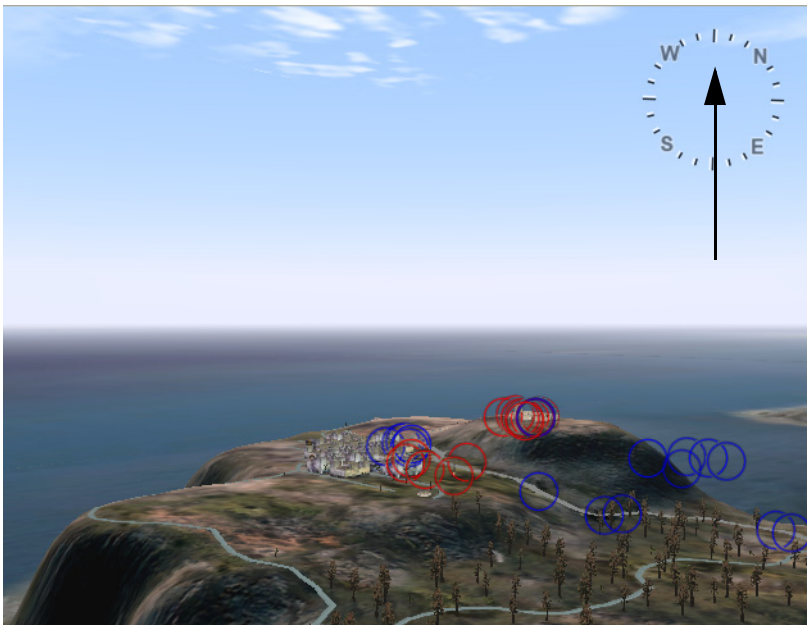


Figure 11-3. Compass

To enable or disable the compass:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 10-2).
3. Select the observer mode for which you want to change the compass setting.
4. In the list of observer settings, select or clear Compass.



You can also enable or disable the compass by choosing **Observer** → **Compass** or clicking the Compass button () on the Observer Settings Toolbar.

11.1.2. Editing the Compass's Attributes

You can configure the compass by editing its schema (ChannelCompass) and model definition (smallCompass and largeCompass). (For details about editing schemas, please see Section 17.2, "Creating and Editing Schemas," in *VR-Forces Configuration Guide*.)

By default, all channels have the largeCompass keyword. To change the style of compass used for a channel, change the keyword in the Keywords attribute for the channel. To completely remove the compass, delete the keyword. For details about setting channel attributes, please see Section 12.4, "Changing a Channel's Attributes," in *VR-Forces Configuration Guide*.

You can change where the compass is displayed in the window by editing the location attribute. The permissible values are top, topRight, right, bottomRight, bottom, bottomLeft, left, and topLeft.

You can change the color of the compass by editing the darkTickColor and lightTick-Color attributes. The color is defined using the OpenGL scheme for specifying RGBA. The first three values are the RGB color values. The fourth value specifies opacity. Each value can be from 0.0 through 1.0.

11.2. Observer Movement Functions and Keyboard Mappings

When you move the observer, you are executing observer movement functions. An observer movement function is a specific way that the observer can move, such as Move Observer Forward or Move Level Observer Forward. (These functions both move the observer forward, but they use different observer coordinate systems. Observer movement coordinate systems are described in "3D Observer Movement Coordinate Systems," on page 11-6.)

The various movement functions are mapped to the keys on the keyboard and to mouse buttons and the mouse wheel. If you use the default key mappings for the default observer modes, then most of the time, the observer will behave exactly as you would expect it to. However for some types of movement, such as forward and back, the observer can move differently depending on the view, the observer coordinate system, or the attach mode. Therefore, VR-Forces provides several key mappings that let you choose which type of observer movement to use. You can change these key mappings and create your own key mappings. (For details about editing key maps, please see Appendix D, *Creating and Editing Key Mappings*, in *VR-Forces Configuration Guide*.) This section describes how observer movement can vary based on the observer coordinate system and the attach mode.

11.2.1. 3D Observer Movement Coordinate Systems

In the 3D view, the observer moves in the context of one of the following coordinate systems:

- Level Observer Coordinates: Movement is with respect to the terrain. The level observer coordinate system is defined by the current up vector (Z), and a plane normal to that vector into which the observer's X and Y axes are projected. (Essentially, this means that the observer's heading is used while its pitch and roll are considered to be 0.0.)

Movement is always with respect to a plane tangent to the earth's surface. However, in a projected terrain database, such as UTM, Lambert Conformal Conic, or flat earth, the plane is fixed, whereas in a geocentric terrain database the up vector (and hence the tangent plan) changes as the observer's position changes, so movement follows the curvature of the earth.

- Observer Coordinates: Movement is with respect to the observer's body coordinate system.
- Entity Level Body Frame Coordinates: Movement is with respect to an entity's body frame, but ignores pitch and roll.
- Entity Body Frame Coordinates: Movement is with respect to an entity's body frame, including pitch and roll.

The practical difference between observer coordinates and level observer coordinates is that in level observer coordinates, if you are looking down at the terrain, when you move forward and backward, you fly over the terrain, looking down. In observer coordinates, when you move forward and backward, you move down towards the terrain or away from it at whatever angle you are oriented towards it.

11.2.2. 2D Observer Coordinate System

When you use the 2D view, the observer uses a variation of level observer coordinates. Movement is restricted to two dimensions, so there are no movement functions for changing the altitude, pitch, or roll. You can zoom in and out in the 2D view, but this does not change the observer's altitude, it changes the field of view and resolution of the terrain image.

11.2.3. The Attached Context and Observer Movement (3D Only)

The key maps control navigation using a particular coordinate system (observer frame or level observer frame). They also support navigation in alternate coordinate systems. The alternate system varies depending on the attached context of the observer. It represents MÄK's opinion of which alternate coordinate system would be most useful given the attach mode that the observer is using.

The Move Attached Context *direction* (forward, back, up, down, left, right) functions move the observer in the context of its attach mode. Movement is modified by attach modes in the following ways:

- Follow mode: movement is in the entity's level body frame.
- Track mode: movement is in the entity's level body frame.
- Mimic mode: movement is in the entity's body frame.
- Tether mode: movement is in level observer frame.
- Unattached: movement is in level observer coordinates.

Figure 11-4 and Figure 11-5 illustrate the differences between moving the observer in an attached context and without an attached context.

Figure 11-4 illustrates movement in entity level body coordinates (Follow and Track modes). The Move Observer Left function (**A** in Observer Frame key map) moves the observer to the left in observer coordinates, that is left relative to the observer's heading. The Move Attached Context Left function (**Shift** + **A**) moves the observer left in the attached context. In entity level body frame, this means the observer moves left relative to the entity's heading.

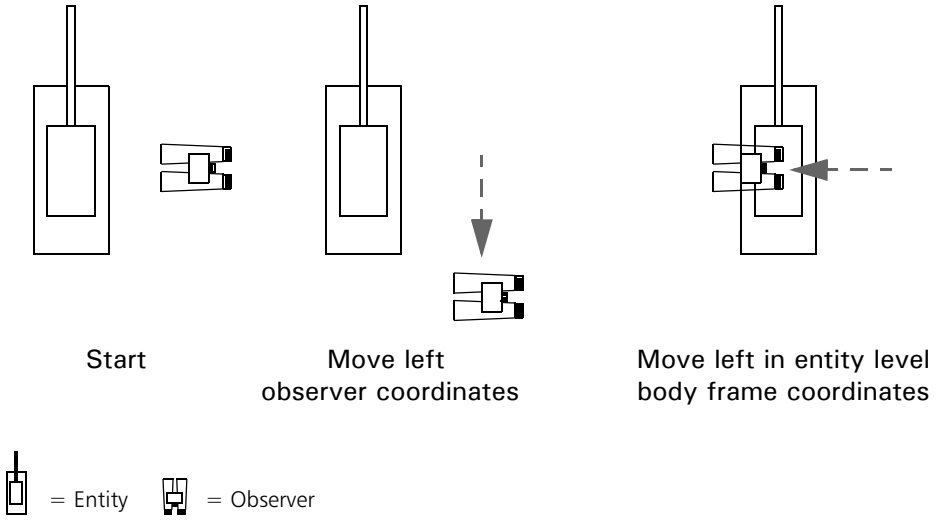


Figure 11-4. Observer movement in entity level body frame coordinates

Figure 11-5 illustrates movement in entity body frame (Mimic mode). The Move Observer Left function moves the observer to the left in observer coordinates, that is left relative to the observer's heading, but ignoring pitch and roll. The Move Attached Context Left function moves the observer left in the attached context. In entity body frame, this means the observer moves left relative to the entity's heading and its pitch and roll.

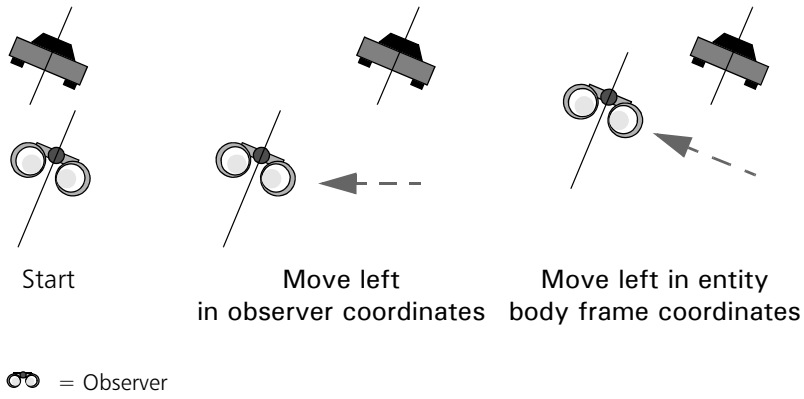


Figure 11-5. Movement in entity body frame coordinates

11.3. Moving the Observer from the Keyboard in the 3D View

VR-Forces maps the observer movement functions to the keyboard using key maps that support the most typical ways that users want to move through the 3D world, either in absolute mode or attached to an entity.

! References to movement key combinations use the key mappings provided with VR-Forces. If you change these key mappings, the instructions may no longer be valid.

As described in "Moving the Observer," on page 11-3, the observer typically moves with respect to the terrain in level observer coordinates or observer coordinates. The corresponding key maps are Level Observer Frame and Observer Frame.

Figure 11-6 illustrates the positional movement keys. Figure 11-7 illustrates the keypad layout for changing orientation and movement speed. For a complete list of movement functions and their keyboard and mouse mappings, please see *VR-Forces Quick Reference Card* or the Key Mapping Editor.

! To use the keypad for navigation, NumLock must be on.



Figure 11-6. Primary movement keys, observer coordinates (3D)

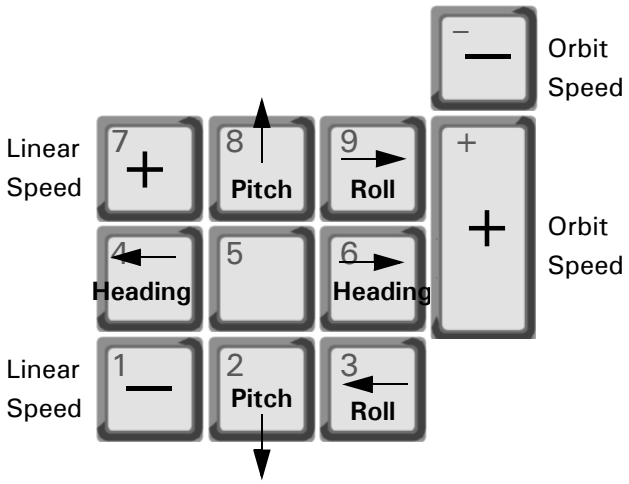


Figure 11-7. Keypad layout, heading, pitch, and roll (3D)

If your keyboard does not have a keypad, as on most laptops, the Laptop key map lets you use the arrow keys to control the left, right, up, and down movement mapped to 4, 6, 8, and 2 on keypads.

➤ To move the observer, press a key combination and hold it for as long as you want the observer to keep moving.

You can press more than one key at a time to combine movement functions. For example, pressing the move forward and move left keys at the same time moves the observer diagonally left and forward.

11.3.1. Navigation Options

This section lists the different ways that you can move the observer using the keyboard. For a complete list of movement functions and their keyboard and mouse mappings, please see *VR-Forces Quick Reference Card*. You can:

- Move forward and backward (observer, level observer, entity, level entity, and attached context coordinates).
- Move up and down (observer, level observer, entity, level entity, and attached context coordinates).
- Move left and right (observer, level observer, entity, level entity, and attached context coordinates).
- Rotate (heading) left and right (observer and level observer coordinates).
- Rotate (pitch) up and down (observer and level observer coordinates).
- Rotate (roll) left and right (observer and level observer coordinates).
- Orbit left and right.
- Orbit up and down.
- Look (move camera) left and right.
- Look (move camera) up and down.
- Tilt (move camera) left and right.
- Reset the observer to the default position and orientation for the attached mode.
- Reset the camera to the default position and orientation.
- Zoom the view.

For details about orbiting, please see "Orbiting," on page 11-12. For details about moving the camera, please see "Moving the Camera," on page 11-12.



When you view a geocentric terrain and move the observer more than 6 million meters away from the earth's surface, the view is forced to focus toward the center of the earth. The practical effect of this mode is that the left and right movement keys orbit the eyepoint around the earth, rather than moving the entire earth to the left or right.

11.3.2. Orbiting

You can orbit the observer (revolve left, right, up, and down) around an attached object or the point on the terrain that is at the center of the window. (If the center of the screen is in the sky, orbit does not work.) When you orbit, the observer's heading continuously changes so that it always faces the point or object around which it orbits. You can change the camera to look at something other than the orbit point while you are orbiting.

To orbit around an object:

1. Attach the observer to the object.
2. Press the right or left arrow keys.

To orbit around a point:

1. If you are attached to an entity, detach (press **`z`**).
2. Press the right or left arrow keys.

11.3.3. Moving the Camera

By default, the observer always looks in the same direction as its heading. However, just as the turret on a tank can swivel and fire in directions other than its body's heading, the observer can look in a direction other than its heading. The ability to look around is enabled in the camera. You can move the observer's camera up, down, left, right, tilt left, and tilt right. (The default key mappings are **`Ctrl`** + *Keypad number*, where *number* is 2, 3, 4, 6, 8, or 9.)

! To use the keypad for navigation, **`NumLock`** must be on.

---

i You can tell that the camera is moving, but the heading is not by observing the compass. If you change the heading, the view of the terrain changes and the compass moves. If you move the camera, but keep the heading static, the view of the terrain changes, but the compass does not move.

➤ To move the camera, press the keys mapped to the camera movement functions.

11.4. Moving the Observer Using the Mouse (3D View)

You can change the observer's position and orientation with the mouse. (To see the full list of mouse mappings, view the Mouse Mappings page on the Application Settings dialog box.)

Using the mouse, you can:

- Drag the terrain.
- Drag the view.
- Orbit the terrain.
- Change orientation.
- Teleport (jump).

11.4.1. Dragging the Terrain

Dragging the terrain combines the move forward, backward, left, and right functions.

To drag the terrain:

1. Click a point on the terrain and hold down the left mouse button.
2. Drag the mouse left, right, forward, and backward.



Dragging the terrain is most effective when you are moving large distances, such as several kilometers. To move small distances, use the keyboard.

Disabling Terrain Dragging

➤ To disable terrain dragging, on the Observer Control Panel, clear the Allow Terrain Dragging check box (Figure 11-8).

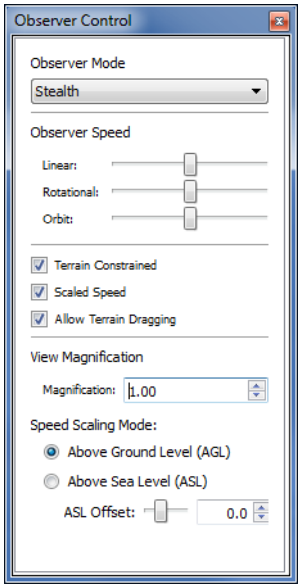


Figure 11-8. Observer Control Panel

11.4.2. Changing the Observer's Orientation with the Mouse

➤ To change the observer's orientation, hold down the right mouse button and drag the mouse. The observer rotates in the direction you drag the mouse.

11.4.3. Dragging the View

Dragging the view combines the observer up, down, left, and right movement functions. Unlike dragging the terrain, which requires you to click a point on the terrain, you can drag the view from any observer location.

➤ To drag the view, hold down the middle mouse button (or wheel) and drag the mouse.

11.4.4. Orbiting the Terrain with the Mouse

➤ To orbit the terrain with the mouse, press **`Ctrl`** and drag the mouse.

11.4.5. Teleporting the Observer

You can teleport the observer to an entity, a prop, or a place on the terrain. Teleporting moves the observer 90% of the way from its current position to the location or object clicked on.

To teleport the observer:

- **`Shift`**+click on the terrain or prop that you want to jump to.
- **`Ctrl`**+**`Shift`**+click on the entity that you want to jump to.

11.5. Moving the Observer in the 2D View

In the 2D view, you move the observer using the mouse and keyboard just as you do in the 3D view. Many of the key and button mappings for the 2D view are the same as those in the 3D view. However, the 2D mappings do not support movement that is not meaningful in two dimensions, such as up and down (altitude), pitch, and roll.



When Orient North is enabled, you cannot change the heading of the observer. For information about Orient North, please see "Forcing Orientation North," on page 11-20.

11.5.1. Moving the Observer from the Keyboard in the 2D View

Figure 11-9 illustrates the positional movement keys for the 2D view. Figure 11-10 illustrates the keypad layout for changing heading and movement speed. For a complete list of movement functions and their keyboard and mouse mappings, please see *VR-Forces Quick Reference Card*.

To use the keypad for navigation, NumLock must be on.



Figure 11-9. Primary movement keys, observer coordinates (2D)

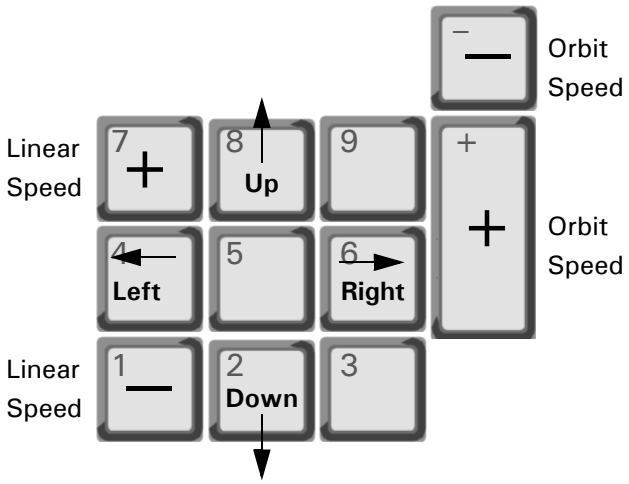


Figure 11-10. Keypad layout, heading, pitch, and roll (2D)

11.5.2. Moving the Observer Using the Mouse in the 2D Projection

You can change the observer's position and heading with the mouse. You can change the observer's position by dragging the terrain, as described in "Dragging the Terrain," on page 11-13. You can change heading by orbiting, as described in the next section. (To see the full list of mouse mappings, view the Mouse Mappings page on the Application Settings dialog box.)

Orbiting the Terrain with the Mouse

When Orient North is enabled, you cannot orbit. For information about Orient North, please see "Forcing Orientation North," on page 11-20.

➤ To orbit the terrain with the mouse, press the right mouse button and drag the mouse.

11.6. Zooming the Observer

You can zoom the observer in the 3D and 2D projections. Although the apparent effect is similar for top-down views, they are conceptually and behaviorally different. In addition to a straight "in-and-out" zoom, you can zoom the view to the extents of an entity or group of entities.

11.6.1. Zooming the Observer in the 3D Projection

If you want to see something at a higher level of detail without moving the observer, you can zoom the observer view. Zooming the observer narrows the field of view and magnifies objects in the view. The magnification level is saved in saved views. 3D zoom has the following characteristics:

- When you zoom the observer view, its movement and mouse look speeds are decreased as the magnification increases, to allow for increased control.
- When you switch between 3D and 2D, VR-Forces calculates a 2D zoom level to match the 3D view. If you then switch back to 3D, it remembers the observer location and zoom level.
- If you zoom in from a very high altitude there may be scene corruption due to insufficient precision in the z-buffer. This can be limited by moving the observer closer to the subject before zooming in. This issue is typically only encountered above 10,000 meters.
- When you zoom, VR-Forces respects the environmental conditions at the observer's location. For example, if the observer is above the cloud layer, zooming does not let you see below the clouds. Or, if it is raining at the observer's location, the visibility will not improve as you zoom in on a distant object.
- After zooming in, when you zoom out, you can only zoom out to the 0 magnification level. That is, you can only zoom out back to the observer's location. If you want to see more of the terrain, you must move the observer to a greater distance from the terrain.
- Some terrains are not designed to be viewed at higher magnification. Performance may decrease if magnification increases the level of detail of large areas of the terrain. For example, suppose you are looking down a line of trees that is in the center of the view. At the normal magnification level there are a certain number of trees at each quality level, say 10 high, 20 medium, and 100 low (for a total of 130 trees). Suppose also that performance will be acceptable only when visualizing 15 or fewer high quality trees. If you magnify the scene 2x, you may now see 20 high, 40 medium, and 70 low quality trees. You see the same number of trees, but the performance will decrease because you are visualizing more high quality models.

➤ To zoom in, press the **=** key, press **`Ctrl`** and scroll the mouse wheel forward, or on the Observer Control Panel specify a value in the Magnification box.

➤ To zoom out, press the **–** (dash) key, press **`Ctrl`** and scroll the mouse wheel backward, or on the Observer Control Panel specify a value in the Magnification box.

---

i Each press of the **=** or **–** key, or increment of the mouse wheel, increases or decreases magnification by a factor of 1.25.

➤ To reset magnification to 1.0, press zero (**`0`**) or press the middle mouse button.

11.6.2. Zooming the Observer in the 2D Projection

In the 2D projection you can zoom the observer view closer to and away from the terrain. However, unlike in 3D, in which you cannot zoom out past the location of the observer, there is essentially no limit on zooming in and out because in 2D, the observer does not actually have altitude.

➤ To zoom in, press **e**, scroll the mouse wheel forward, or to step the zoom, press the **=** key.

➤ To zoom out, press **q**, scroll the mouse wheel backward, or to step the zoom, press the **–** (dash) key.

➤ To reset the observer to the default zoom level, press the **Space bar**.

11.6.3. Setting the Scale Factor for Paged LODs

When you magnify the view, all distances used for determining LODs (including paged LODs) are scaled by the view magnification. In some cases, this is more aggressive than desired for the paged LODs. To remedy this problem, you can change the amount by which the paged LOD distance is scaled relative to the view magnification. For example, if view magnification is 8.0 and the Paged LOD Magnification Scale Factor is 2.0, then the distance used for paging would be scaled by 4.0 (8.0/2.0) rather than by the full factor of 8.0.

To set the scale factor for paged LODs:

1. Choose **Settings** → **Experimental Features**. The Experimental Features dialog box opens.
2. Adjust the Paged LOD Magnification Scale Factor slider.

11.6.4. Zooming to Extents

You can zoom the observer to the extents of the terrain or selected entities. Zoom to extents works as follows:

- If no entities are selected and the observer is not attached to an entity, the observer view is reset to the terrain center. This is the same behavior as pressing **Space bar**.
- If the observer is attached to an entity, the observer resets to the default attached view. This is the same behavior as pressing **Space bar**.
- If entities are selected and the observer is not attached to an entity, the observer view is redisplayed to show the extents of the selected entities. In the 3D view, this is a top-down view.

Zooming the Observer to the Extents of Selected Entities

You can quickly change the observer view to show the extents of a selected entity or entities. In 2D the observer zooms to the extents. In 3D the observer moves to the correct altitude to view the extents of the entities while looking down at them.

To zoom the observer to the extents of selected entities:

1. Select the entities that you want to zoom in on. To zoom to the extents of all entities, do not select any entities.
2. Choose **View** → **Zoom to Extents**.

Zooming to Entity Extents when a Scenario Loads

You can configure a scenario so that it zooms to the extents of all entities when the scenario loads. This can be convenient when you have entities located in one part of a large terrain.



If a scenario has a saved observer view that is designated as the initial view for the scenario, it overrides Zoom to Scenario Extents on Load.

To zoom to entity extents when a scenario loads:

1. Open the scenario that you want to zoom to extents on load.
2. Choose **Settings** → **Application**. The Application Settings dialog box opens.
3. Select the Session Options page (Figure 5-3).
4. Select the Zoom to Scenario Extents on Load check box.
5. Save the scenario. The next time you load the scenario, if the Zoom to Scenario Extents on Load check box is selected, it will zoom to the entity extents.

11.7. Centering the Observer on an Entity

You can center the view on an entity. When you center the observer view on an entity, the observer attaches to the entity in Tether mode. For details about Tether mode, please see "Tether Mode," on page 12-11.

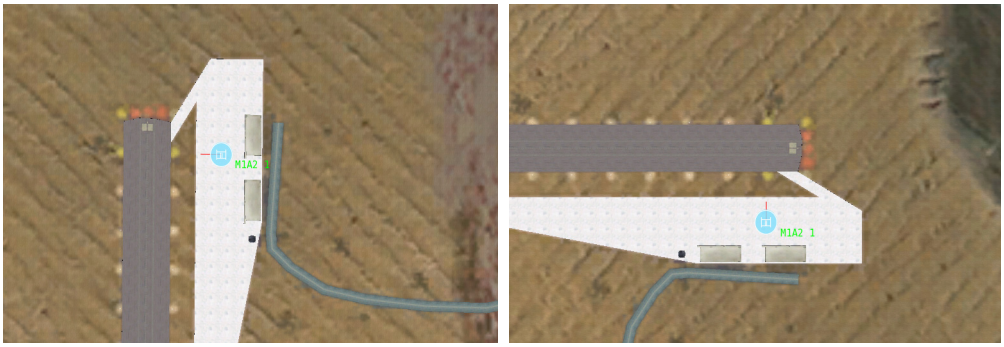
To center the observer on an entity:

1. In the Object Console Summary Panel, right-click a message from the entity you want to center on. A menu is displayed.
2. Choose Center Entity.

11.8. Forcing Orientation North

In the 2D view, by default the observer's heading is forced to stay North (up), regardless of the heading of the entity to which it is attached. You can disable this setting, in which case if you attach to an entity, the view of the terrain rotates to match the heading of the attached entity.

Figure 11-11 illustrates the view when the observer is attached to an entity with orientation forced to North or matched to that of the entity.



Orient North | Orient to entity heading

Figure 11-11. Forcing orientation North

To enable or disable a northward orientation:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 8-6).
3. In the Observer Mode list, select Plan View.
4. In the list of observer settings, select or clear Orient North.



You can also enable or disable northward orientation by choosing **Observer** → **Orient North** or clicking the Orient North button () on the Observer Settings Toolbar.

11.9. Enabling and Disabling View Constraints (3D Only)

View constraints control how the observer behaves when it approaches solid objects in the view, such as the ground, buildings, and vegetation. When constraints are enabled, the observer has a 1 meter bounding sphere that prevents it from moving through terrain or props.



If view constraints are enabled, you may not be able to move the observer into buildings to view and place entities. Changing the near clipping plane or transparency will let you see through walls, but in constrained mode, you will still not be able to move the observer through the walls.

To enable or disable view constraints, use one of the following methods:

- Press **/** (slash).
- Choose **Observer** → **Constrain to Terrain**.
- On the Observer Control Panel (Figure 11-8), select or clear the Terrain Constrained check box.
- Click the Constrain to Terrain button () on the Observer Settings Toolbar.

11.10. Controlling Navigation Speed

When you navigate through the terrain, there are times you want to move quickly, covering large amounts of terrain, and times you want to move slowly, such as when you are moving down a street or within a building. VR-Forces has the following options for controlling navigation speed:

- Set the navigation speed manually. The observer always moves at the specified speed.
- Use speed scaling to automatically move at different speeds under different circumstances. You can change the rate at which speed is scaled.

In either case, the controls for changing speed are the same. Their effect varies depending on whether or not speed scaling is enabled.

11.10.1. Enabling or Disabling Speed Scaling

When speed scaling is enabled, VR-Forces makes the following assumptions:

- If the observer is unattached, the higher above the terrain it is, the faster you want to move.
- The closer you are to an attached object, the more slowly you want to move.

To enable or disable speed scaling, use any of the following methods:

- On the Observer Control Panel (Figure 11-8), select or clear the Scaled Speed check box.
- Press the multiply key (*) (not the asterisk key).
- Click the Speed Scaling button () on the Observer Settings Toolbar.

Setting the Speed Scaling Mode

When speed scaling is enabled, VR-Forces scales observer speed based on the its height above the terrain. Even if the observer is higher than all terrain points, speed can still be affected if there are changes to the terrain, such as mountains. This can be annoying if it happens unexpectedly. Therefore, you can choose to base speed scaling on height above ground level, or height above sea level. In that latter case, changes to the terrain would not affect the speed at which the observer moves. If you set speed scaling to use height above sea level, you can specify an offset.

To set the speed scaling mode, use one of the following methods:

- On the Observer Control Panel (Figure 11-8), set the Speed Scaling Mode to Above Ground Level (AGL) or Above Sea Level (ASL)
- On the Display Settings dialog box, Observer Settings page, set the Speed Scaling Mode to Above Ground Level (AGL) or Above Sea Level (ASL)

11.10.2. Changing the Observer's Movement Speed

You can change the observer's linear, rotational, and orbital speed. If speed scaling is enabled, these controls change the speed gain. If speed scaling is disabled, the controls change the absolute speed.

To change the observer's speed, use any of the following methods:

- Adjust one of the Observer Speed sliders on the Observer Control Panel (Figure 11-8) or on the Observer Settings page on the Display Settings dialog box. The default (center of the slider) is 1. The range is 0.1 to 10.
- Press one of the following keys to change linear speed:
  - **`Keypad 7`** (faster) or **`1`** (slower).
  - **`Ctrl`** + mouse wheel forward or back.
  - Movement key + mouse wheel forward or back.
- Press **`Keypad +`** (faster) or **`-`** (slower) to change orbital speed.



When you change the observer's speed using the keyboard or mouse, if the Observer Control Panel or the Observer Settings page is open, you can see the scale slider moving to the new setting.

11.11. Saving and Recalling Views

At any given moment, the state of the observer is called its view. You can capture views so that you can quickly return to them. You can also save them for future use. VR-Forces automatically saves observer views as part of the scenario (*scenario_name.osrx*). You can also save views independently of scenarios. The procedures in this section describe that process.

If the observer is not attached to an entity, its view is its position and orientation in the simulated environment. If the observer is attached to an entity, a view saves the following information:

- Attach mode.
- Positional offset.
- Marking text.



When you save a view in the 2D view, the X and Y values are saved. The zoom level is not saved.

Saved views are managed in the Observer Views Panel (Figure 11-12).



Figure 11-12. Observer Views Panel

Its buttons (as indicated by the callouts) have the following meanings:

- Save the current view (1).
- Mark view to load on scenario startup (2).
- Delete the selected view (3).
- Save all views to a file (4).
- Save the selected view to a file (5).
- Load a saved views file and append it to the list (6).
- Load a saved views file and replace the current list (7).
- Clear the list (8).

Saved views are keyed to the marking text of the object to which the observer is attached.

To save the current view:

1. Choose **View** → **Observer Views Panel**. The Observer Views Panel opens (Figure 11-12).
2. Click the Add button (+). The current view is added to the list.

Views are remembered only while VR-Forces remains running. To save a view for future use, you must save it to a file. Please see "Saving Views to a File," on page 11-26.

11.11.1. Recalling a View

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

Animating Transitions Between Observer Views

Normally, when you select an observer view, the observer jumps immediately to the new view. However, VR-Forces 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 Observer Views Panel that you want the observer to move to.



If the observer view that you select uses a different observer mode (for example, from PVD mode to Stealth), VR-Forces does not change the observer mode. The view moves to an approximation of the the desired view in the current observer mode.

11.11.2. Deleting Views

To delete a view:

1. In the Observer Views Panel, select the view you want to delete.
2. Click the Delete button ().

Clearing the View List

➤ To remove all views from the list, in the Observer Views Panel, click the Clear List button ().

11.11.3. Specifying the Startup View

If a scenario has observer views, you can specify that one of them be the view selected when the scenario loads. (The observer view does not get selected when you rewind a scenario.)

To specify a view as the startup view:

1. Select the view.
2. Click the star button () on the Observer Views Panel.

11.11.4. Saving Views to a File

You can save views to a file independently of the scenario and load them at a later time. You can save an individual view or all of the views in the list.

The default location for saved views is *./appData/settings/vrfGui*.

To save the selected view to a file:

1. In the Observer Views Panel, click the Save Individual View button ( ). The Save Selected Observer dialog box opens.
2. Select the file to which you want to save the view, or type the name of a new file.
3. Click OK.

To save all views to a file:

1. In the Observer Views Panel, click the Save All Views button ( ). The Save Observer Views dialog box opens.
2. Select the file to which you want to save the views, or type the name of a new file.
3. Click OK.

11.11.5. Loading Views

If a scenario has observer views saved as part of the scenario, they are loaded automatically when you load the scenario. If you have saved views independently of the scenario, you can load them and append them to the list in the Observer Views Panel, or you can replace the list in the panel with the saved views.

To load views and append them to the list of views:

1. In the Observer Views Panel (Figure 11-12), click the Load Views and Merge button ( ). The Open Observer Views dialog box opens.
2. Select the saved views file that you want to load.
3. Click OK.

To load views and replace the list of views:

1. In the Observer Views Panel, click the Load Views and Replace button ( ). The Open Observer Views dialog box opens.
2. Select the saved views file that you want to load.
3. Click OK.

11.12. Controlling the Observer from Other Applications

The VR-Forces front-end can respond to view control messages sent by applications such as the MÄK Logger. It also responds to view control messages sent from entity plans and global plans. You can write VR-Link-based applications to send view control messages to VR-Forces using the classes in the VR-Vantage Control Toolkit. You can use view control messages to:

- Specify the attach mode and attached entity.
- Set the observer's position and orientation.

Acceptance of view control messages is a global setting.



The view control PDUs are experimental, and are not an official part of DIS. They are of PDU kind 202.

11.12.1. Enabling the Processing of View Control Messages

➤ To enable or disable the processing of view control messages, choose **Settings** → **View Controls**, or click the View Control button () on the Display Settings Toolbar.



12. Attaching the Observer to Objects

This chapter describes attachment modes and how to attach the observer to entities and props.

Attaching the Observer to an Entity or Prop 12-2
Attaching to an Articulated Part 12-4
Attaching the Observer in Mimic Track or Tether Track Mode 12-4
Filtering the Object Attachment List 12-4
Detaching from an Entity 12-5
Hiding the Attached Model (3D Projection Only) 12-5
Attaching to Props 12-5
Improving Performance when Attaching to Ground Entities 12-5
Attach Modes in the 3D View 12-6
Absolute Mode 12-6
Follow Mode 12-7
Mimic Mode 12-8
Mimic Track Mode 12-9
Space Follow Mode 12-10
Tether Mode 12-11
Tether Track Mode 12-12
Track Mode 12-13
Attach Modes in the 2D View 12-14
Attaching the Observer at Startup 12-14

12.1. Attaching the Observer to an Entity or Prop

You can attach the observer to an entity and follow it as it moves through the simulation. You can also attach the observer to props.



This procedure applies to Follow, Tether, Track, Mimic, and Space Follow modes. To attach in Tether Track or Mimic Track mode, please see "Attaching the Observer in Mimic Track or Tether Track Mode," on page 12-4.

To attach the observer to an entity or prop, do one of the following:

- Right-click an object in the Objects List Panel (Figure 12-1) or on the terrain and select **Attach** *mode*, where *mode* is one of the supported attach modes. (For details about attach modes, please see "Attach Modes in the 3D View," on page 12-6 and "Attach Modes in the 2D View," on page 12-14.)

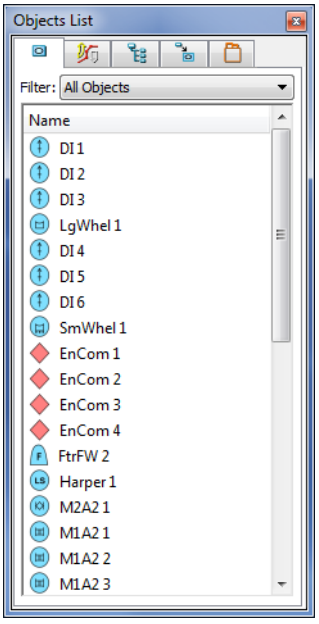


Figure 12-1. Objects List Panel

- Select an entity or prop. (For details about selecting entities and props, please see "Selecting Entities, Tactical Graphics, and Props," on page 6-4.) On the Attachments Panel (Figure 12-2), or on the Attach Object Toolbar (Figure 12-3), click the Attach button ( ) to attach with the current attachment mode.
- On the Attachments Panel or Attach Object Toolbar, click the next or previous arrows to attach to the next (or previous) object in the object list.

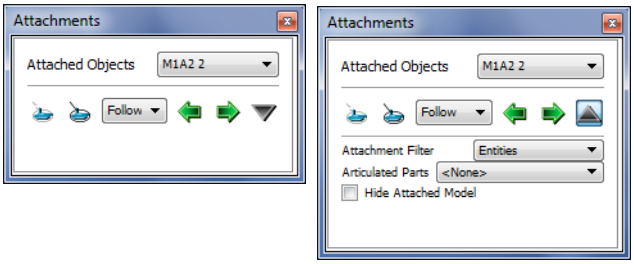


Figure 12-2. Attachments Panel



Figure 12-3. Attach Object Toolbar

- Press greater than (**>**) to attach to the next entity or prop in the object list without regard to the attachment filter. Press less than (**<**) to attach to the previous object.
- Press period (**.**) to attach to the next entity or prop allowed by the attachment filter. Press comma (**,**) to attach to the previous object.

- When you select the next or previous object, VR-Forces chooses next and previous based on the order in which the objects were discovered. This may or may not represent the order in which they are listed in the Objects List Panel.
- When you move the attachment from one object to the next, VR-Forces preserves whatever changes you have made to the observer's view relative to the attached object. So if, for example, you moved the observer so that it is not looking at the attached object, when you attach to the next entity, the observer will not be looking at this object. It will be in the same relative position to the object that you previously established. To quickly return to the default offset for the attachment mode, press **`Space bar`**.

12.1.1. Attaching to an Articulated Part

You can attach the observer to an articulated part on an entity, such as a turret. When you attach to an articulated part, the observer moves with the articulated part instead of the main body of the entity.

To attach to an articulated part:

1. Attach the observer to an entity, as described in "Attaching the Observer to an Entity or Prop," on page 12-2.
2. On the Attachments Panel (Figure 12-2), click the down arrow. The Attachments Panel expands.
3. In the Articulated Parts list, select the part to which you want to attach the observer.

12.1.2. Attaching the Observer in Mimic Track or Tether Track Mode

To attach the observer in Tether Track and Mimic Track modes, you must specify a primary attached entity and a secondary entity (the tracked entity).

To attach the observer in Mimic Track mode or Tether Track mode:

1. Right-click the primary entity in the Objects List Panel or in the window. A popup menu is displayed.
2. Choose **Attach Mimic Track** or **Attach Tether Track**. The cursor changes to show an arrow and a Plus sign.
3. Click the secondary entity in the Objects List or in the window.

12.1.3. Filtering the Object Attachment List

Although you can attach to entities and props, it is likely that at any given time you will only want to attach to one type of object (usually entities). To reduce the number of objects that you have to iterate through when using Attachments Panel arrows to pick the next or previous object, you can filter the attachment list.



The filter for buoys and beacons applies to S-57 feature data, not buoys that are simulated entities, such as sonobuoys.

To filter the attachment list:

1. On the Attachments Panel (Figure 12-2), click the down arrow. The Attachments Panel expands.
2. In the Attachment Filter list, select the type of scene element that you want to attach to.

12.1.4. Detaching from an Entity

➤ To detach the observer from the entity to which it is attached, on the Attachments Panel (Figure 12-2), click the Detach button (), or press **z**.

12.1.5. Hiding the Attached Model (3D Projection Only)

You can hide the model for the entity to which the observer is attached.

To hide the attached model:

1. On the Attachments Panel (Figure 12-2), click the down arrow. The Attachments Panel expands.
2. Select the Hide Attached Model check box.

12.1.6. Attaching to Props

Since props do not move, attaching the observer to a prop does not result in any observer movement. The principal benefit of attaching to a prop is to let you quickly refocus the observer on a particular location in the scene. If you attach in Track mode, using the keyboard movement keys allows you to orbit around the prop.

12.1.7. Improving Performance when Attaching to Ground Entities

When the observer is attached to air entities, Z-fighting can affect visual quality. To prevent this problem, VR-Forces needs to calculate the entity's height above the terrain. However, the calculation can affect performance. Unfortunately, when this calculation is enabled (the default), it must be applied to all entity types. If you know that you will want to attach the observer to ground entities, but not air entities (or if you do not care about Z-fighting), you can disable the height above terrain calculation, which may improve performance when the observer is attached to entities. To disable the height above terrain calculation, start VR-Forces with the `-S | --suppressHatIntersectOnAttach` command line option.

12.2. Attach Modes in the 3D View

Attach modes are combinations of position and orientation models and constraints that produce a useful behavior for the observer. Except for Absolute mode, attach modes describe how movement of the entity to which the observer is attached affects the position and orientation of the observer.

VR-Forces supports the following attach modes:

- Absolute: The observer is not attached to an entity; it moves in space independently of all entities.
- Follow: The observer position and heading changes with the entity's position and heading.
- Mimic: The observer position and orientation changes with the entity's position and orientation.
- Mimic Track: The observer's position changes with the entity's position while its orientation is focused on a second entity.
- Space Follow: The observer is a fixed distance from the entity and the entity model is scaled to be observable from a great distance.
- Tether: The observer's position changes with the entity's position.
- Tether Track: The observer's position changes with the entity's position while its heading is focused on a second entity.
- Track: The observer's heading changes to track the entity's position and heading.

The attach modes are described in detail in the following sections. For details about how to attach the observer to an entity, please see "Attaching the Observer to an Entity or Prop," on page 12-2.

12.2.1. Absolute Mode

In Absolute, or free-fly mode, the observer is not attached to an entity. Movement of entities does not affect the observer in any way.

12.2.2. Follow Mode

In Follow mode, the observer maintains its relationship to the attached entity as follows:

- When the entity's position changes, the observer's position changes to maintain its offset.
- When the entity's heading changes, the observer's heading and position change to maintain the offset.
- When the entity's pitch or roll change, the observer pitch and roll do not change.

Figure 12-4 illustrates the characteristics of Follow mode.

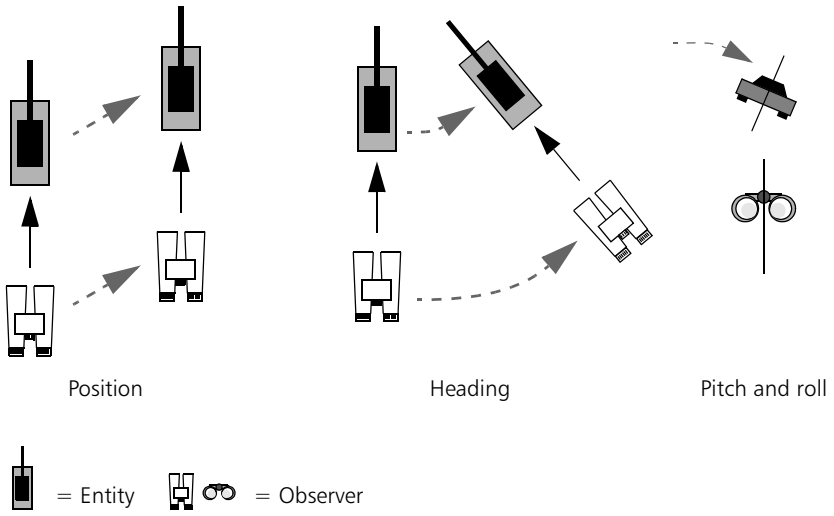


Figure 12-4. Follow mode

12.2.3. Mimic Mode

In Mimic mode, the observer behaves as though it is rigidly connected to the entity. Use Mimic mode to create the effect of riding in a vehicle cockpit, or of following a vehicle from an external location while connected to it by a solid, invisible rod. The observer maintains its relationship to the attached entity is as follows:

- When the entity's position changes, the observer's position changes to maintain its offset.
- When the entity's heading changes, the observer's heading and position change to maintain the offset.
- When the entity's pitch or roll change, the observer's pitch and roll change and the observer's position changes to maintain its offset.



When cockpit displays are enabled, you cannot change the position or orientation of the observer relative to the entity.

Figure 12-5 illustrates the characteristics of Mimic mode.

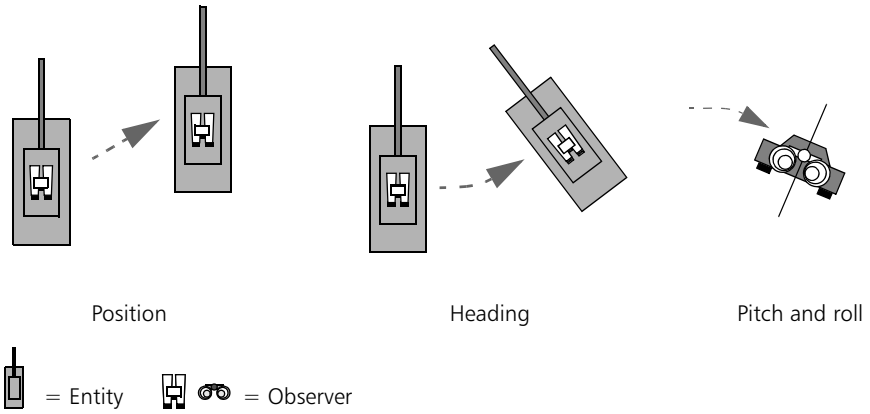


Figure 12-5. Mimic mode

12.2.4. Mimic Track Mode

Mimic Track mode combines Mimic mode and Track mode. The observer's position changes with that of the entity to which it is attached, as in Mimic mode, while its orientation changes to remain focused on a secondary, tracked entity.

Orientation controls are disabled.

Figure 12-6 illustrates the characteristics of Mimic Track mode.

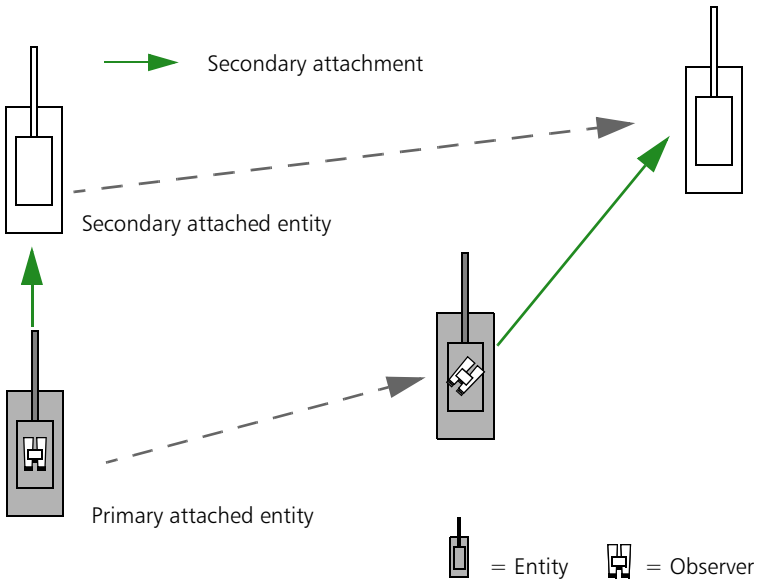


Figure 12-6. Mimic Track mode

12.2.5. Space Follow Mode

Space Follow mode attaches the observer above an entity, looking down at it. Although it can be used with any entity, its primary purpose is for attaching to space objects such as satellites. You can also usefully use it with fixed-wing and rotary-wing entities.

The observer is attached at a fixed distance, determined by the near clipping plane for the object. For a satellite, this distance is approximately 500,000 meters. Since it is not possible to see regular models at that distance, the attached object is scaled to be visible. (You cannot change the scaling of the attached object.) Other objects are not scaled (except to the extent that model scaling might be enabled).

The observer maintains its relationship to the attached entity as follows:

- When the entity's position changes, the observer's position changes to maintain its offset.
- When the entity's heading changes, the observer's heading changes.
- When the entity's pitch or roll changes, the observer's pitch or roll does not change.

You cannot move the observer towards or away from the attached entity. You can rotate the observer around the entity, but you cannot go below the horizontal plane of the entity.

Figure 12-7 illustrates the characteristics of Space Follow mode.

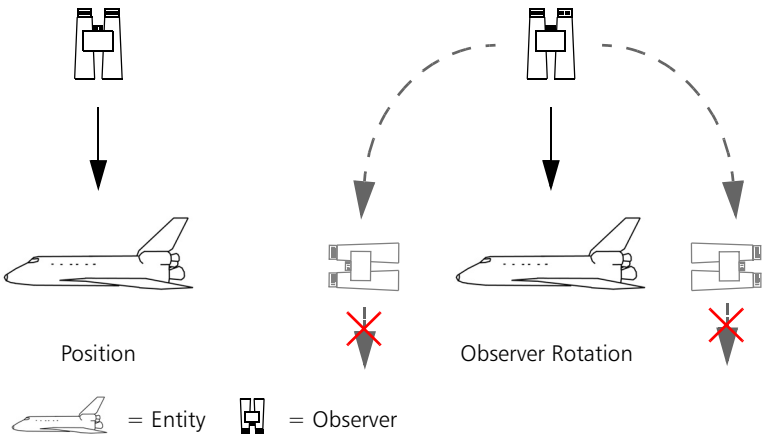


Figure 12-7. Space Follow mode

12.2.6. Tether Mode

In Tether mode, the observer's maintains a constant offset from the attached entity, but it is not affected by the entity's heading, pitch, or roll, as follows:

- When the attached entity's position changes, the observer's position changes to maintain its offset.
- When the attached entity's heading changes, the observer's heading does not change.
- When the attached entity's pitch or roll change, the observer's pitch and roll do not change.

Figure 12-8 illustrates the characteristics of Tether mode.

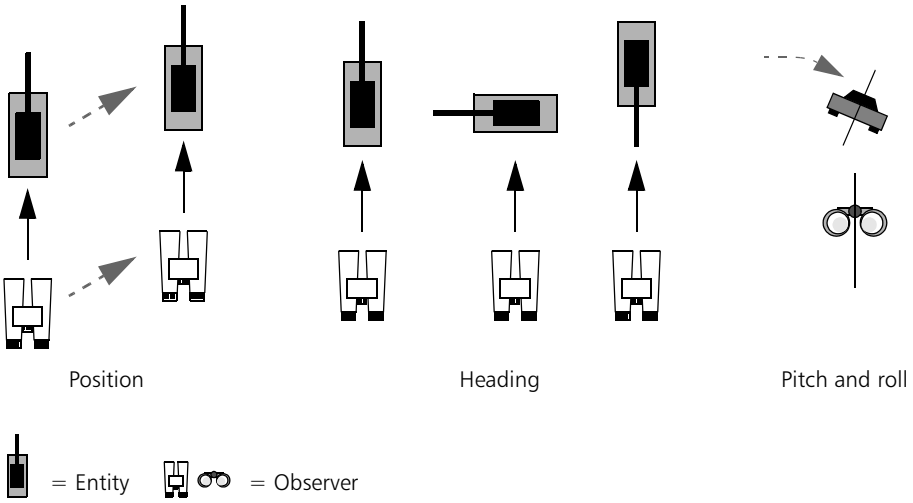


Figure 12-8. Tether mode

12.2.7. Tether Track Mode

Tether Track mode combines Tether mode and Track mode. The observer's position changes with that of the entity to which it is attached, as in Tether mode, while its orientation changes to remain focused on a secondary, tracked entity.

Orientation controls are disabled.

Figure 12-9 illustrates the characteristics of Tether Track mode.

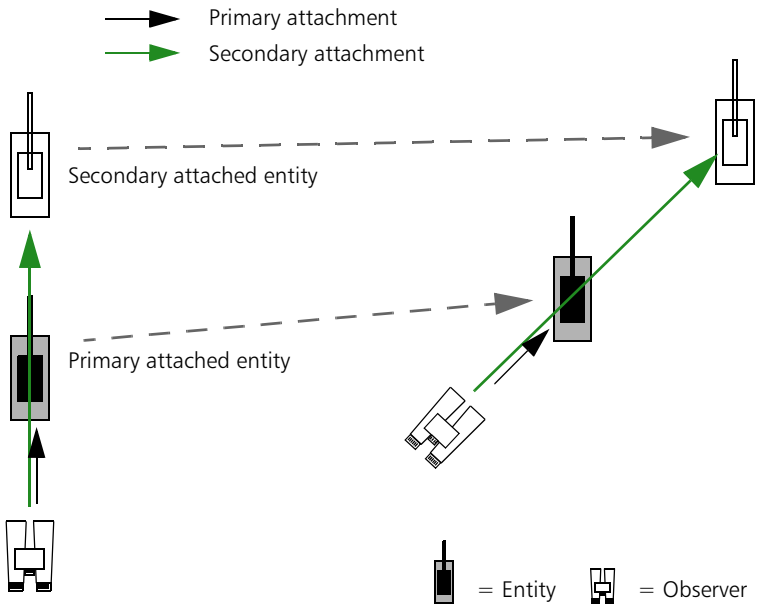


Figure 12-9. Tether Track mode

12.2.8. Track Mode

In Track mode, the observer's heading changes to track the attached entity's movement, keeping it centered in the field of view. The observer's position does not change. The observer maintains its relationship to the attached entity as follows:

- When the attached entity's position changes, the observer's heading changes to keep the entity centered in the observer's view; the observer's position does not change.
- When the attached entity's heading changes, the observer's heading does not change.
- When the attached entity's pitch or roll change, the observer's pitch and roll do not change.

Figure 12-10 illustrates the characteristics of Track mode.

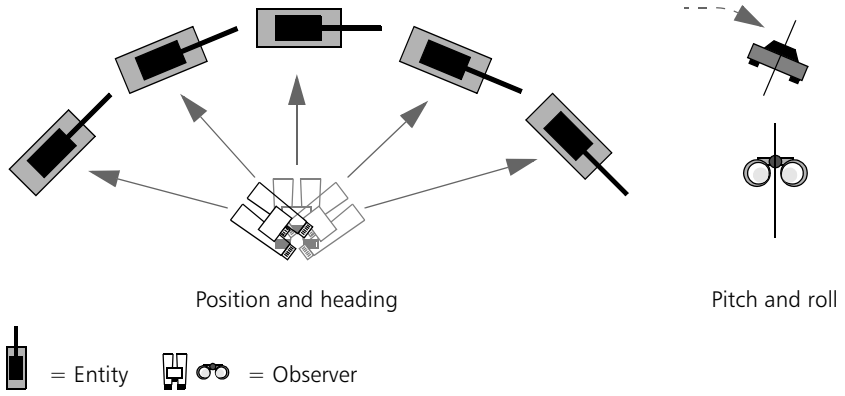


Figure 12-10. Track mode

12.3. Attach Modes in the 2D View

The 2D view supports Follow mode and Tether mode. In Follow mode, the heading of the observer changes with the entity (if Orient North is disabled); in Tether mode, it does not. You can use the standard movement keys (**w**, **a**, **s**, **d**) to offset the observer with respect to the attached entity's position. You can use the orbit keys (left arrow, right arrow) to offset the observer's orientation with respect to the entity's orientation (in Follow mode) or the terrain.

If the observer is attached to an entity in the 3D view and you switch to the 2D view, the attach mode is mapped to either Follow or Tether modes, as shown in Table 12-1. If you then switch back to the 3D view the original attach mode is restored.

Table 12-1: 3D to 2D attachment mode mapping

3D Attach Mode	2D Attach Mode
Follow	Follow
Mimic	Follow
Mimic Track	Follow
Space Follow	Follow
Tether	Tether
Tether Track	Tether
Track	Tether

12.4. Attaching the Observer at Startup

To attach the observer to an entity when you load a scenario, save a view and specify it as a favorite, as described in "Specifying the Startup View," on page 11-25.



13. Using Terrain Profiles

This chapter explains what terrain profiles are and how to display and configure them.

Terrain Profiles 13-2
Displaying a Line's Terrain Profile 13-3
Configuring Terrain Profiles 13-3
Analyzing Terrain Using a Terrain Profile Line 13-5

13.1. Terrain Profiles

You can display terrain profile windows for line objects that have height (in other words, routes and lines, but not phase lines), for intervisibility lines, and for an entity's track history. You can also create lines specifically for the purpose of showing the terrain profile. A terrain profile displays:

- A graph of the line against the terrain.
- Entities within a specified distance of the line or track.
- Location of any one point along the line.
- The location on the grid of the cursor. This helps you estimate distances between the grid lines.

In Figure 13-1, the yellow line (1) represents the terrain and the red line (2) shows a route. The location of the cursor in grid coordinates is displayed (3) and the location of one point along the terrain is displayed (4). Several entities are close to the route and they are shown as well (5).

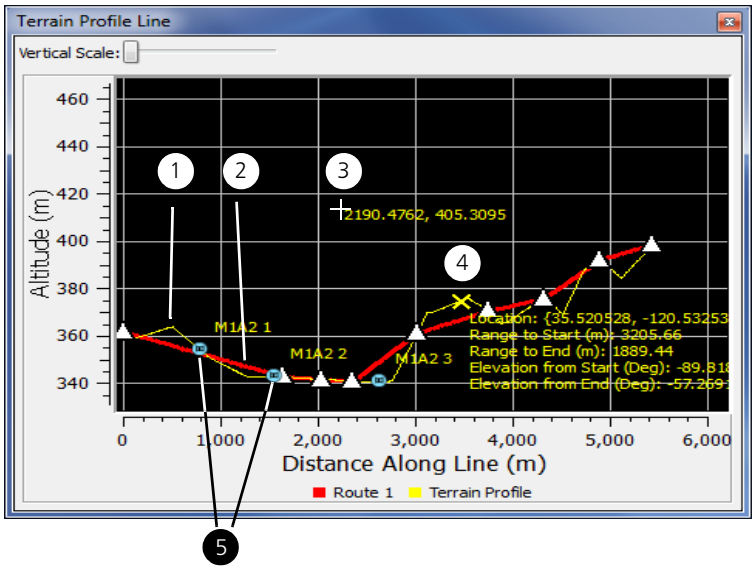


Figure 13-1. Terrain profile

13.1.1. Displaying a Line's Terrain Profile

You can display a terrain profile for routes, intervisibility lines, and terrain profile lines.

To view a line's terrain profile, use one of the following methods:

- Right-click a line and choose **Terrain Profile** from the context menu.
- Open the Edit dialog box for a line and click Terrain Profile. (You can configure terrain profiles to open automatically when you open an Edit dialog box.)

To display the terrain profile for a track history, follow the procedure in "Displaying a Terrain Profile for a Track History," on page 8-13.

13.1.2. Configuring Terrain Profiles

You can configure the following aspects of a terrain profile:

- The frequency, in distance, at which terrain height is sampled to create the graph in the Terrain Profile window.
- The maximum number of points to sample along a line.
- Whether or not to show entities that are close to the line and how close to the line entities need to be to be shown.
- Whether or not to show the vertices of the line.
- Whether or not terrain profiles should be displayed automatically when a line is created or edited.

To specify the frequency with which terrain height is sampled:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Terrain Profile Settings page (Figure 13-2).

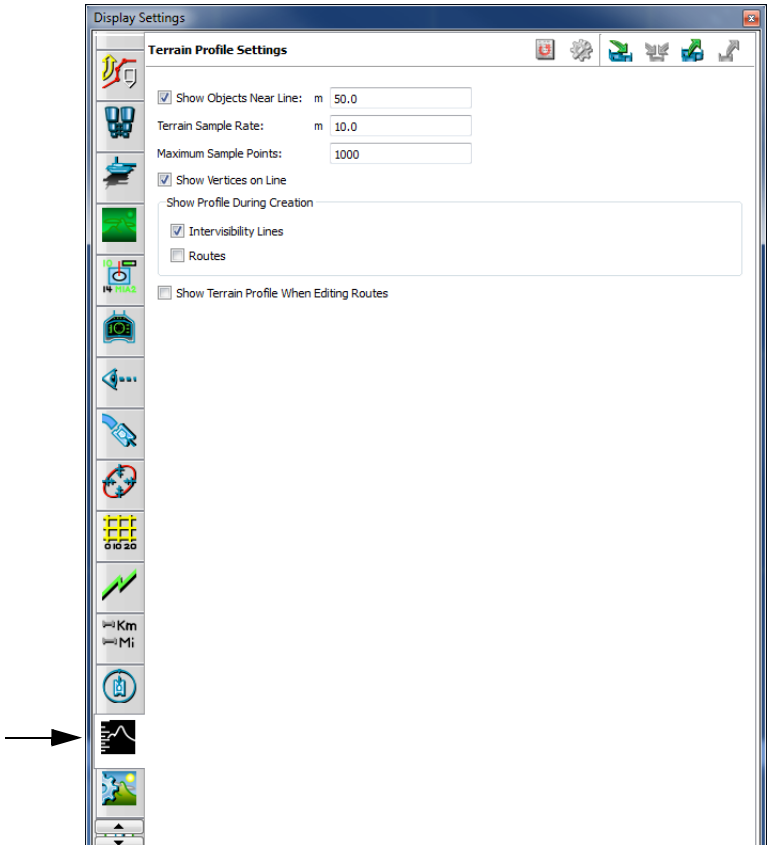


Figure 13-2. Terrain Profile Settings page

3. If you want terrain profiles to show entities that are near the line, select the Show Objects Near Line check box.
4. To specify how far from a line entities can be to be displayed in a terrain profile, change the value in the Show Objects Near Line text box.
5. To specify how frequently the terrain is sampled, type a value in the Terrain Sample Rate box.
6. To change the maximum number of sample points for a line, edit the Maximum Sample Points value.
7. If you want to show the vertices of the line in a terrain profile, select the Show Vertices On Line check box.
8. If you want a terrain profile window to open when you create a line, select the line types for which you want to show a terrain profile in the Show Profile During Creation group box.
9. If you want a terrain profile window to open when you edit a line, select Show Terrain Profile When Editing Routes.

13.2. Analyzing Terrain Using a Terrain Profile Line

A terrain profile line is a special line used to analyze the terrain. It is more versatile than a point-to-point intervisibility line because it can have multiple points, which allows a finer analysis of the terrain. Unlike a route (for which you can also view terrain profiles), a terrain profile line is not published, so it has less overhead and does not clutter up object lists.

To create a terrain profile line:

1. Choose **Intervisibility** → **Create Terrain Profile Line**. A Terrain Profile window opens and the cursor changes to draw mode.
2. Click on the terrain to place the vertices of the line. As you add vertices, the Terrain Profile window updates to show the profile of the line (Figure 13-3).

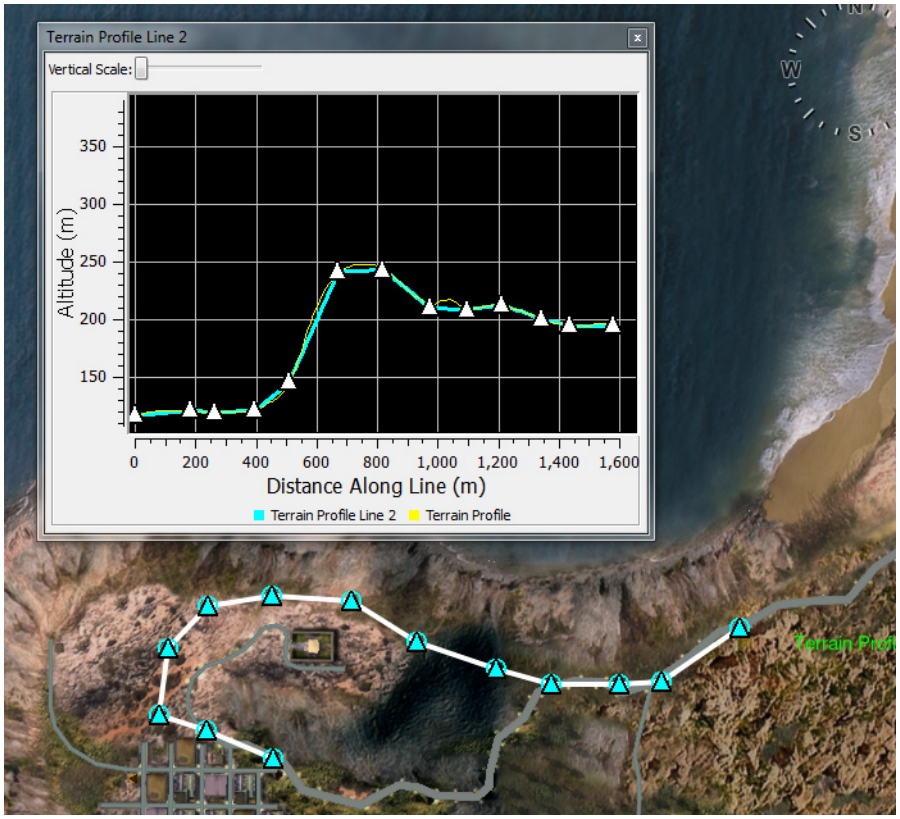


Figure 13-3. Terrain profile line

3. Right click to finish the line.

After you create a terrain profile line, you can edit it just like any other line. For details, please see Section 6.5.1, "Editing Vertices," in *VR-Forces Scenario Management Guide*.



14. Intervisibility

This chapter explains how to use the intervisibility feature. For an introduction to intervisibility, please see "Intervisibility (Line-of-Sight)," on page 3-38.

Types of Intervisibility 14-2
- Point-to-Point Intervisibility 14-2
- Intervisibility Fan 14-3
- Entity Intervisibility 14-3

Intervisibility Objects can be Transient or Permanent 14-4
- Creating a Transient Intervisibility Object 14-5
- Creating a Permanent Intervisibility Object 14-6

Editing an Intervisibility Object 14-7

Configuring Intervisibility Lines and Fans 14-8

Pinning an Intervisibility Object to an Entity 14-9

Showing and Hiding Intervisibility Lines 14-9

Displaying Intervisibility Line Information 14-10

Displaying an Intervisibility Line's Terrain Profile 14-11

Deleting Intervisibility Objects 14-11

14.1. Types of Intervisibility

VR-Forces supports the following types of intervisibility:

- Point-to-point (intervisibility line).
- Intervisibility fan.
- Entity intervisibility.

The intervisibility fan and entity intervisibility are displayed as a circular area. However, intervisibility calculations are made within a cylindrical area extending above and below the terrain, so that intervisibility calculations can be made for air platform entities and subsurface entities.

In the 2D view, intervisibility lines are fully visible on the terrain. In the 3D view, they may pass through the terrain.

14.1.1. Point-to-Point Intervisibility

Point-to-point intervisibility (Figure 14-1) shows the line-of-sight between any two points. The portion of the line that is visible from the starting point is green. The portion that is not visible is red. You can change the colors used.

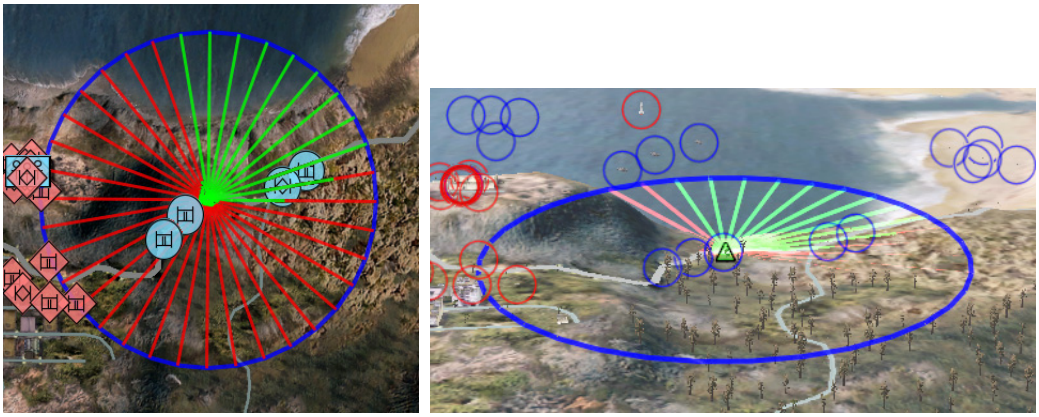


2D 3D

Figure 14-1. Point-to-point intervisibility

14.1.2. Intervisibility Fan

An intervisibility fan (Figure 14-2) shows a series of lines radiating from the center of a circle to its outer edge. Each ray shows intervisibility along its path from the origin to the border. You can configure the distance between rays. (For details, please see "Configuring Intervisibility Lines and Fans," on page 14-8.)



2D 3D

Figure 14-2. Intervisibility fan

14.1.3. Entity Intervisibility

Entity intervisibility (Figure 14-3) shows which entities are visible from the center of a defined area and whether or not the entities can see the center point. If you center the intervisibility circle on a point on the terrain, you can see which entities can see that point. If you lock the circle to an entity, you can see which entities can see that entity and which it can see. (For details, please see "Configuring Intervisibility Lines and Fans," on page 14-8.)



2D 3D

Figure 14-3. Entity intervisibility

14.2. Intervisibility Objects can be Transient or Permanent

You can configure intervisibility objects to be transient or permanent. (For details, please see "Configuring Intervisibility Lines and Fans," on page 14-8.) A transient intervisibility object displays intervisibility from the origin point (defined by a mouse click) to wherever you move the cursor. If you click the mouse again, you reset the origin. You cannot use the mouse for other actions until you exit intervisibility mode (by right-clicking). This mode is convenient when you want to quickly check intervisibility, but do not want to keep intervisibility objects in the display.

A permanent intervisibility object stays in the display after you create it. You can resize the object, edit drawing options, and move it around the display. Intervisibility objects are not saved as part of the scenario.

14.2.1. Creating a Transient Intervisibility Object

Transient intervisibility objects do not stay in the display after you quit intervisibility mode.

To display transient intervisibility:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Intervisibility Settings page (Figure 14-4).

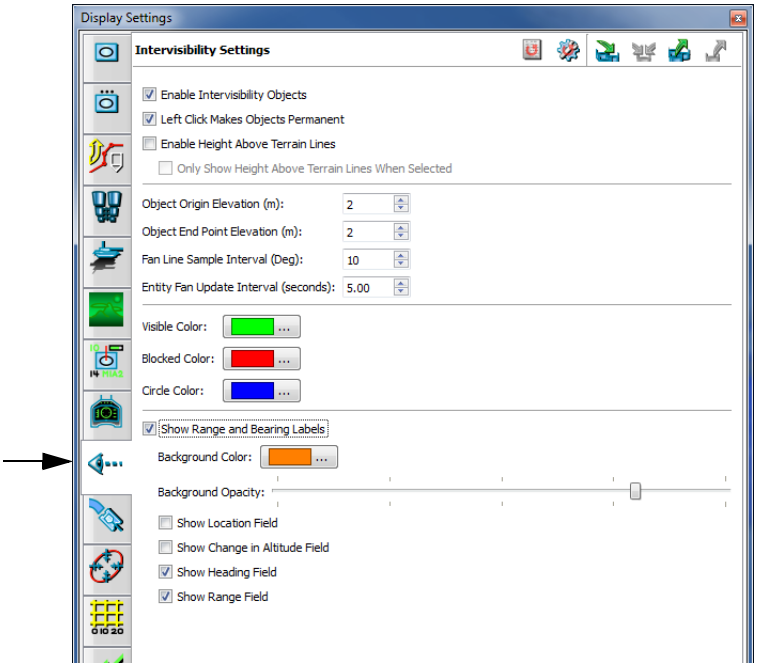


Figure 14-4. Intervisibility Settings page

3. Clear the Left Click Makes Objects Permanent check box.
4. Choose **Intervisibility** → **Create *intervisibility_type***, where *intervisibility_type* is a type of intervisibility object, or click one of the buttons on the Intervisibility Toolbar.
5. Click on the terrain to place the origin of the intervisibility object.
6. Move the cursor around the terrain to expand and contract the object. For point-to-point intervisibility, moving the cursor also changes the direction of the line.
7. To change the origin of the intervisibility display, click another location on the terrain.
8. Right-click to cancel intervisibility mode.

14.2.2. Creating a Permanent Intervisibility Object

Permanent intervisibility objects remain in the display after you create them. (However, they are not saved as part of the scenario.)

To create a permanent intervisibility object:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Intervisibility Settings page (Figure 14-4).
3. Select Enable Intervisibility Objects.
4. Select Left Click Makes Objects Permanent.
5. Choose **Intervisibility** → **Create *intervisibility_type***, where *intervisibility_type* is a type of intervisibility object, or click one of the buttons on the Intervisibility Toolbar.
6. Click on an entity or on a location on the terrain where you want the origin of the intervisibility object to be located. An intervisibility object is displayed. As you move the cursor closer to the origin, or farther away from it, the circle or line contracts and expands. If you are displaying point-to-point, you can rotate the line around the origin before you specify the end point.
7. Click on the terrain where you want to fix the end point of a line or border of a circle.

14.3. Editing an Intervisibility Object

You can change the following aspects of an intervisibility object:

- Change the location of either end point of an intervisibility line, to change its length and direction.
- Change the origin of an intervisibility fan, thereby moving the entire fan.
- Change the diameter of an intervisibility fan.



If an intervisibility object is anchored to an entity, you cannot change the origin.

To change the length or diameter of an intervisibility object:

1. To edit a line, double-click its end point.

   To edit a fan, select the object, then double-click the outer ring of the fan.

   The cursor changes (✣).
2. Drag the end point or circle towards or away from the origin. If you are editing an intervisibility line, you can move the end point to any location.
3. Click to exit edit mode.



To revert to the original size of the intervisibility object, right-click while you are still in edit mode.

To move an intervisibility fan or line:

1. Double-click a fan's center point or any place on a line. The cursor changes.
2. Drag the object to a new location.
3. Click to set the location.



To revert to the original location of the intervisibility object, right-click while you are still in edit mode.

14.4. Configuring Intervisibility Lines and Fans

You can configure the following attributes of intervisibility objects:

- The elevation of the origin and end point.
- The sampling rate for intervisibility fans.
- The update rate for entity intervisibility fans.
- Display of height above terrain lines (3D) and height labels (3D and 2D).
- The colors used to display intervisibility objects.
- Display of range and bearing labels. For more information about range and bearing labels, please see "Displaying Intervisibility Line Information," on page 14-10.

When you change colors or the sampling rate, changes affect all existing intervisibility objects. Changes to the origin and end point elevation only affect newly created objects.

To configure intervisibility objects:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Intervisibility Settings page (Figure 14-4).
3. Change the settings as desired. Table 14-1 describes the settings.

Table 14-1: Intervisibility settings

Option	Description
Enable Height Above Terrain Lines	Enables or disables display of height above terrain lines (3D) and height labels (3D and 2D).
Show Height Above Terrain Lines Only for Selected	If selected, HAT lines are displayed only for the selected intervisibility objects.
Object Origin Elevation	The height, in meters, above the terrain at the origin point from which to calculate intervisibility.
Object End Point Elevation	The height, in meters, above the terrain at the end point to use to calculate intervisibility.
Fan Line Sample Interval	The distance, in degrees, between each ray in a intervisibility fan. After changing this value, press **`Enter`** or click away from the field.
Entity Fan Update Interval	How frequently, in seconds, to update the entity intervisibility fan.
Visible Color Blocked Color	The colors to use for the portion of an intervisibility line that an entity can see and the portion where visibility is blocked.
Circle Color	The color to use for the outer circle of an intervisibility fan.
Show Range and Bearing Labels	Enable or disable display of range and bearing labels.

Table 14-1: Intervisibility settings

Option	Description
Background Color	Specify the background color for range and bearing labels.
Background Opacity	Specify the opacity of the background for range and bearing labels.
Show Location Field	If range and bearing labels are displayed, show the location of the endpoints.
Show Change in Altitude Field	If range and bearing labels are displayed, show the difference in altitude between the endpoints.
Show Heading Field	If range and bearing labels are displayed, show the heading of each endpoint relative to the other point.
Show Range Field	If range and bearing labels are displayed, show the length of the intervisibility line.

14.5. Pinning an Intervisibility Object to an Entity

When you pin an intervisibility object to an entity, the intervisibility object moves with the entity and you know that its origin is centered on the entity.

To pin an intervisibility object to an entity:

1. Select the entity to which you want to pin the intervisibility object.
2. Choose **Intervisibility** → **Create *intervisibility_type***, where *intervisibility_type* is a type of intervisibility object, or click one of the buttons on the Intervisibility Toolbar. The origin is placed on the entity icon or model.
3. Specify the end point for the intervisibility object.

14.6. Showing and Hiding Intervisibility Lines

You can toggle the display of intervisibility lines on and off.

To enable or disable the display of intervisibility lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Intervisibility Settings page (Figure 14-4).
3. Select or clear the Enable Intervisibility Objects check box.



You can also enable and disable display of intervisibility objects by choosing **Intervisibility** → **Enable/Disable Intervisibility**, or by clicking the Enable/Disable Intervisibility button () on the Intervisibility Toolbar.

14.7. Displaying Intervisibility Line Information

You can display the following information about an intervisibility line:

- Location of each endpoint.
- Heading of the line from each endpoint.
- Altitude of each endpoint.
- Length of the line (range).



Figure 14-5. Intervisibility line information (2D)

The information label has a background. You can adjust the color and opacity of the background.

To display information about an intervisibility line:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Intervisibility Settings page (Figure 14-4).
3. Select or clear the Show Range and Bearing Labels check box.
4. Select the check boxes for the type of information you want to display. For descriptions of the check box options, please see Table 14-1.
5. To change the background color for the label:
   a. Click the Background Color button. A color picker opens.
   b. Select the color you want to use for the background.
   c. Click OK.
6. To change the opacity of the background, adjust the Background Opacity slider.

14.8. Displaying an Intervisibility Line's Terrain Profile

By default, a terrain profile window is displayed automatically when you create an intervisibility line. The terrain profile can be very helpful when you try to understand how terrain changes are affecting intervisibility. Figure 14-6 shows an obstructed intervisibility line and its terrain profile, whose graph illustrates the line intersecting the terrain.

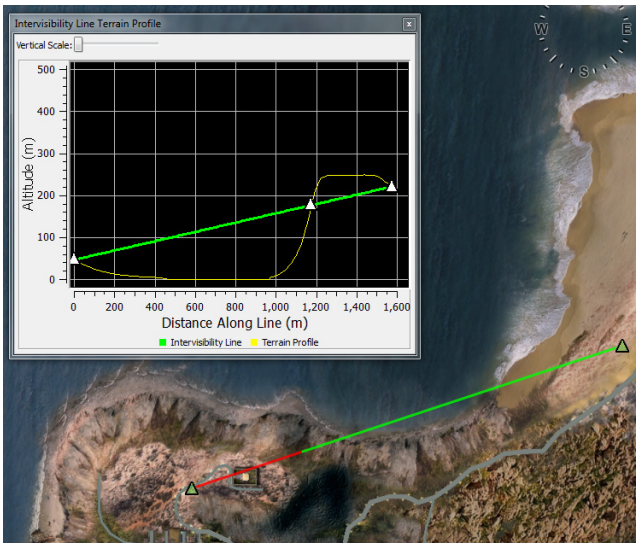


Figure 14-6. Terrain profile for an intervisibility line

You can disable this setting on the Display Settings dialog box, Terrain Profile Settings page. For details, please see "Configuring Terrain Profiles," on page 13-3.

14.9. Deleting Intervisibility Objects

Before you delete an intervisibility object, note the following:

- If you delete an entity that has an intervisibility object pinned to it, the intervisibility object also gets deleted.
- If an intervisibility object is pinned to an entity, you can delete the intervisibility object without affecting the entity by selecting the outer circle or end point and deleting it.

➤ To delete an intervisibility object, select the object and choose **Intervisibility** → **Delete**.



15. Sound Effects

VR-Forces can play sound effects for entities, fire events, and detonation events.

Introduction to Sound Effects 15-2
Enabling Sound Effects 15-2
Setting the Sound Volume 15-3
Specifying the Range in which Sounds are Heard 15-3
Associating Sound Files with Entities and Events 15-4
Mapping a Sound File to an Entity 15-5
Mapping Sound Files to Fire and Detonation Events 15-8

15.1. Introduction to Sound Effects

VR-Forces 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 sounds will be heard only when the observer is within a certain distance of the entity or event that generates them.

Sound effects are not supported on Linux.

15.2. Enabling Sound Effects

To enable sound effects:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Audio Settings page (Figure 15-1).

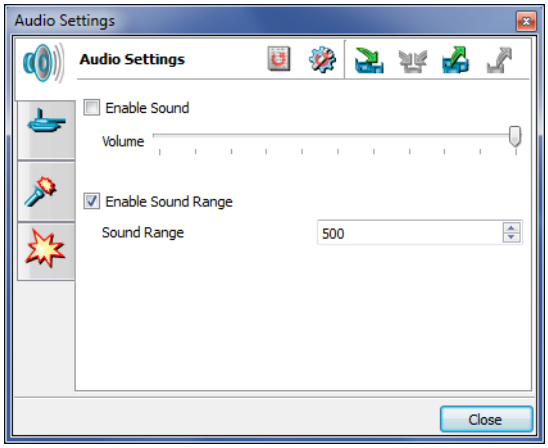


Figure 15-1. Audio Settings page

3. Select the Enable Sound check box.

You can also enable and disable sound effects by clicking the Audio button (▾) on the Display Settings toolbar.

15.2.1. Setting the Sound Volume

You can adjust the volume of sound effects.

To set the sound volume:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Audio Settings page (Figure 15-1).
3. Adjust the Volume slider.



You can also change the volume by clicking the arrow on the Audio button ( ) (on the Display Settings toolbar) and adjusting the Audio Volume slider.

15.2.2. Specifying the Range in which Sounds are Heard

You can specify that sound effects be played only if the observer is within a certain distance of the entities that can emit sounds. If sound range is enabled, as soon as the observer moves beyond the range from an entity, sound effects are shut off. If this option is disabled, sound effects are played for all entities in the scenario regardless of their distance from the observer. However, even if sound range is not enabled, as the distance between the observer and entities increases, sound levels drop off.

To specify the range in which sounds can be heard:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Audio Settings page (Figure 15-1).
3. Select the Enable Sound Range check box.
4. Specify a value in the Sound Range text box.

15.3. Associating Sound Files with Entities and Events

VR-Forces includes a set of sound files (WAV format) for generic entity platforms and for fire and detonation events. The files are in *./data/Audio/Sounds*. Sound files are associated with entity type enumerations. You can change the mappings and add new mappings.

You add and edit mappings between entity types, fire events, or detonation events and sound files on pages in the Audio Settings dialog box. You can add or edit mappings using mapping dialog boxes or directly in the mappings windows. The mapping dialog boxes behave differently depending on the state of the mapping window when you open the dialog box, as follows:

- If you click the Add Mapping button and have not selected a mapping in the Entity Mapping Settings window, you are presented with a default entity type (all wildcards).
- If you click the Add Mapping button and have selected an existing mapping, the dialog box displays the mapping you selected. From here, you can edit the existing mapping or create a new one.

When you edit a mapping, the result depends on whether you edit the entity or event enumeration or the sound file. The differences are based on the requirements for uniqueness in a mapping. The uniqueness of an entity type or fire event mapping is based on the entity type enumeration. The uniqueness of a detonation event mapping is based on the entity type enumeration and the detonation result value. Table 15-1 lists the results of different editing actions.

Table 15-1: Editing sound mapping files

Action	Result
Edit enumeration in dialog box.	New mapping is added. The previous mapping is not changed.
Edit enumeration in place.	New mapping replaces previous mapping.
Edit sound file in dialog box or in window.	Mapping to current entity or event is updated.
Edit channel in dialog box or in window.	Mapping to current entity or event is updated.
Edit detonation result in dialog box.	New mapping is added. The previous mapping is not changed.
Edit detonation result in window.	New mapping replaces previous mapping.

15.3.1. Mapping a Sound File to an Entity

An entity type enumeration can only have one sound file mapped to it. A sound file can be mapped to as many entity types as you want.

Adding a New Entity Sound Mapping

To add a new entity sound mapping:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Entity Sound Editor page (Figure 15-2).

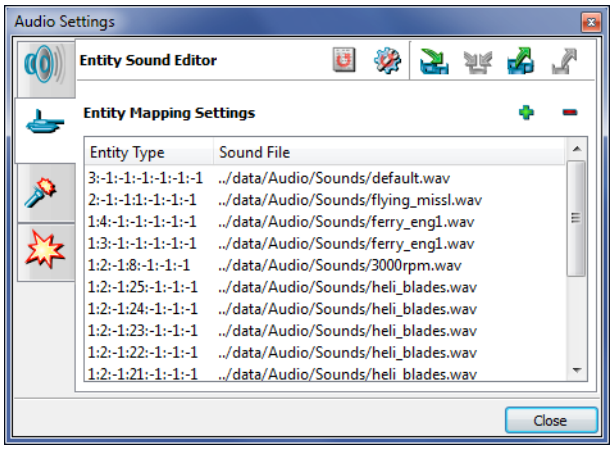


Figure 15-2. Entity Sound Editor page

3. Click the Add button (+). The Create New Entity Type Sound Mapping dialog box opens (Figure 15-3).

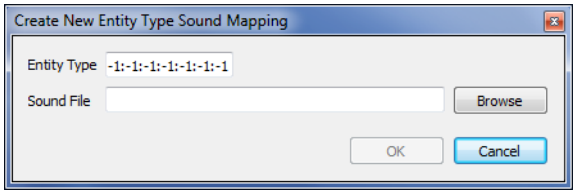


Figure 15-3. Create New Entity Type Sound Mapping dialog box



You cannot open the Create New Entity Type Sound Mapping dialog box with default options if one of the existing mappings is selected. If you have selected a mapping and want to return to the unselected state, click on white space in the mappings window or select one of the other tabs and then select the Entity Sound Editor tab again.

4. In the Entity Type box, type the enumeration for this mapping.

5. In the Sound File text box, type the pathname of a sound file or click Browse and select a file in the file chooser dialog box.
6. Click OK.



The OK button does not become enabled until you specify a valid entity type and sound file name.

Creating an Entity Sound Mapping from an Existing Mapping

When you edit a mapping, you can either change the sound file associated with the current entity type or create a new mapping with a different entity type.

To create a sound mapping from an existing mapping:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Entity Sound Editor page (Figure 15-2).
3. Select the mapping that you want to use as the starting point.
4. Click the Add button (). The Create Entity Type Sound Mapping from Existing Mapping dialog box opens (Figure 15-4).

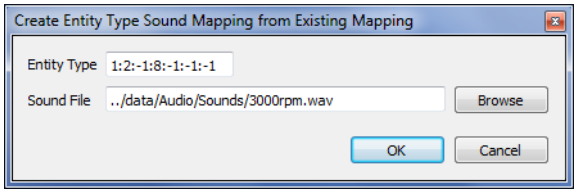


Figure 15-4. Create Entity Type Sound Mapping from Existing Mapping dialog box

5. To change the sound file associated with the entity type, type a new pathname or click Browse and select a sound file.
6. To create a new mapping, change the Entity Type enumeration.
7. Click OK. If you just changed the sound file, the mapping is updated. If you changed the entity type, a new mapping is added to the list.

Editing the Entity Type in the Mappings Window

If you edit the entity type in the mappings window (that is, without using a dialog box), it replaces the previous mapping.

To edit the entity type in the window:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Entity Sound Editor page (Figure 15-2).
3. Double-click the entity type you want to edit. It becomes editable (Figure 15-5).

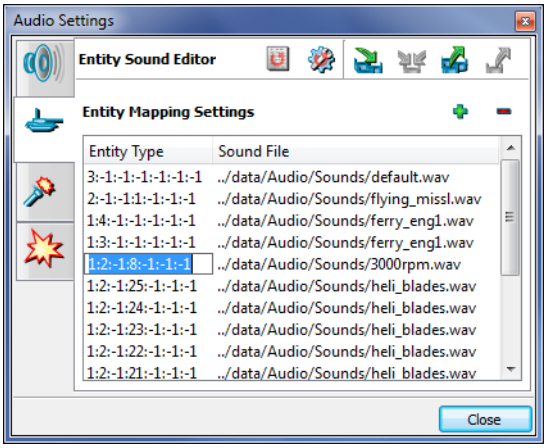


Figure 15-5. Editable entity type

4. Change the entity type enumeration.
5. Press **`Enter`**.

Editing the Sound Mapping in the Mappings Window

When you edit the sound mapping in the mappings window, the change does not affect any other mapping.

To edit the sound mapping in the window:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Entity Sound Editor page (Figure 15-2).
3. Double-click the sound mapping that you want to change. The Sound File dialog box opens.
4. Select the new sound file.
5. Click Open.

15.3.2. Mapping Sound Files to Fire and Detonation Events

When you map a sound file to a weapon fire or a detonation event, you can specify a channel for the sound. When you map a detonation sound, you can specify the detonation result for the mapping.

Specifying a channel for a sound event lets you control the number of sounds that can play concurrently. Sounds that are mapped to channel 0 can all play at the same time. Sounds mapped to any other channel can play only if no other event is using the channel. For example, suppose you map two weapon fire events to channel 1. One of the weapon type fires and the sound is generated. Then the second type of weapon event occurs. Since the first event is using channel 1, the second sound is not generated.

The DIS Enumerations document specifies a list of detonation results, such as Entity Impact, Ground Impact, and Air Hit. The detonation result is part of the detonation message sent over the network. You can specify that a detonation sound be played only for a certain type of detonation result. If you set the result to "any" it will play for any type of detonation. If, for a particular detonation type, you map one sound to "any" and another sound to a specific detonation result, the specific detonation result takes precedence.



You can change fire and detonation mapping settings directly, rather than using the sound mapping dialog box, just like you can change entity type sound mappings. For details, see the sub-procedures under "Mapping a Sound File to an Entity," on page 15-5.

To map a sound file to a weapon fire or detonation:

1. Choose **Settings** → **Audio Settings**. The Audio Settings dialog box opens.
2. Select the Fire Sound Editor page or Detonation Sound Editor page (Figure 15-6), as appropriate.

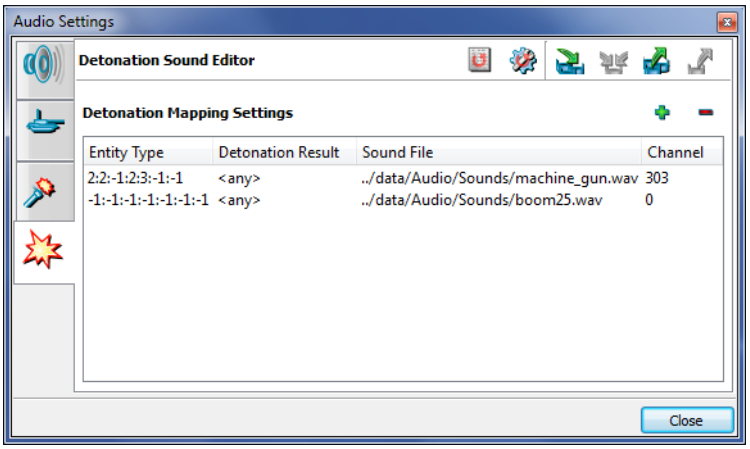


Figure 15-6. Detonation Sound Editor page

3. Click the Add button (+). The Create New Detonation (or Weapon Fire) Sound Mapping dialog box opens (Figure 15-7).

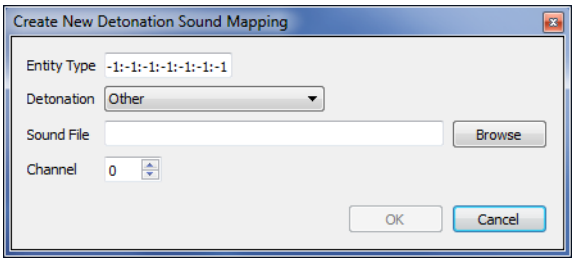


Figure 15-7. Create New Detonation Sound Mapping dialog box

When you click the Add button, if you have selected a mapping in the list, the Create New Sound Mapping from Existing dialog box displays the values for the selected mapping. If you change the entity type or detonation result, a new mapping is added with the new values.

4. In the Entity Type box, type the enumeration for this mapping.
5. Optionally, for detonation sounds, select an option on the Detonation list.
6. In the Sound File text box, type a pathname for the sound file you want to use or click Browse and select a file in the file choose dialog box.
7. Optionally, specify a channel in the Channel box.
8. Click OK.



16. Using Sensors

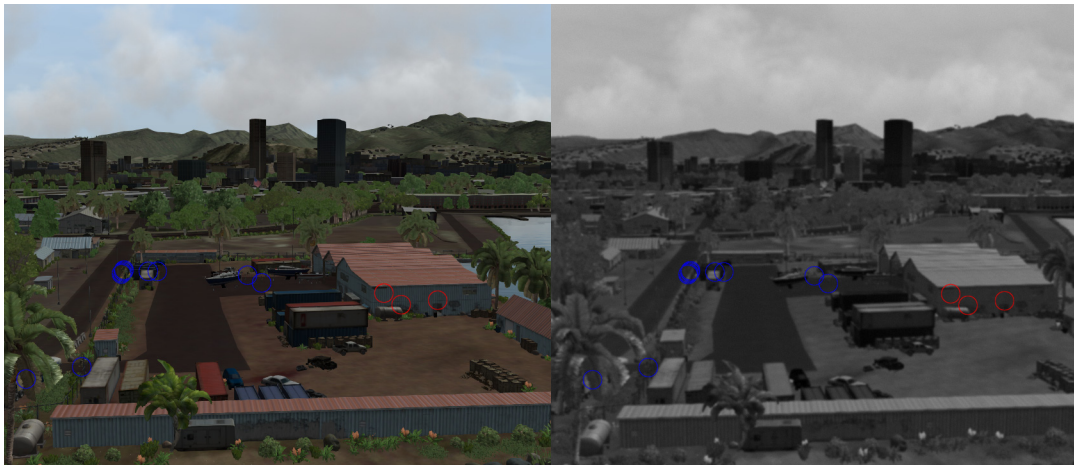
This chapter explains how to enable and configure simulated night-vision goggle (NVG) and infrared views of the scene.

Using Sensors 16-2
Enabling Sensors 16-3
Configuring Sensors 16-4
Adding New Sensors 16-6

16.1. Using Sensors

VR-Forces can simulate the way that the 3D scene would look if an observer were using visual sensor devices such as night vision goggles or viewing the infrared spectrum. This is an observer-specific setting. You can adjust the contrast, blur, and noise of the view.

VR-Forces includes one sensor module – CameraFX, which uses the SenSim libraries from JRM Technologies. The sensors do not take into account the materials of the objects that you are viewing. They simply filter the view to produce the desired effect.



No sensor (daylight) IR sensor

Figure 16-1. CameraFX



The sensors described in this chapter topic are strictly visual effects. They have no relationship to the sensor systems used by entities.

16.2. Enabling Sensors

Sensors are observer-specific (Figure 16-2).

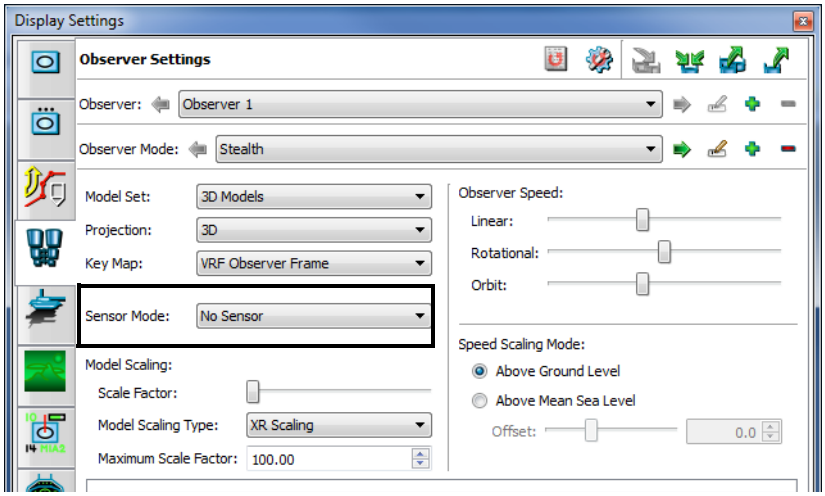


Figure 16-2. Sensor setting for observer mode

VR-Forces includes observer modes that have sensors enabled. The names of the observer modes match the sensor modes that they enable.

- NVG (CameraFX).
- IR (CameraFX).
- IR BlackHot (CameraFX)
- Camera (CameraFX). This mode uses the Visual Camera sensor mode.

The default settings for the Camera observer mode make it look very similar to Stealth observer mode. The difference is that when you use the Camera observer mode, you can change the view by changing the sensor settings, such as blur, noise, and gain.



If you have SensorFX installed, when you run it, the observer modes include the text (SensorFX). The presence of CameraFX or SensorFX in the observer mode name makes it clear which type of sensor you are using.

You can also specify use of a sensor configuration as a channel attribute.

To enable sensors, do one of the following:

- Select an observer mode that has a sensor configured.
- Edit an observer mode to use a sensor.
- Create a new observer mode and configure it with a sensor.
- Change the Sensor attribute for a channel.

For details about creating and editing observer modes, please see "Creating and Editing Observer Modes," on page 10-2. For details about editing channel attributes, please see Section 12.4, "Changing a Channel's Attributes," in *VR-Forces Configuration Guide*.

16.3. Configuring Sensors

You can configure the following aspects of sensors:

- Blur. The sharpness of the image.
- Noise. The graininess of the image.
- Gain. The contrast and brightness of the image. (IR only)

You can configure a separate set of settings for each sensor type. VR-Forces remembers the settings. When you change the settings for a sensor type, they affect the display only if that sensor type is in use. For example, if you select NVG in the sensor list on the Observer Settings page, and select IR in the Sensors list on the Sensor Settings page, the changes you make to the IR configuration do not affect the NVG display you are looking at. If you then switch to IR in the VR-Forces window, the new IR settings take effect. (This probably sounds obvious, but the Sensor Settings page does not automatically display the configuration for the sensor that is being used. If you want to edit the sensor that you are currently using, you have to be sure that you have selected it to edit.)

To configure sensors:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Sensor Settings page (Figure 16-3).

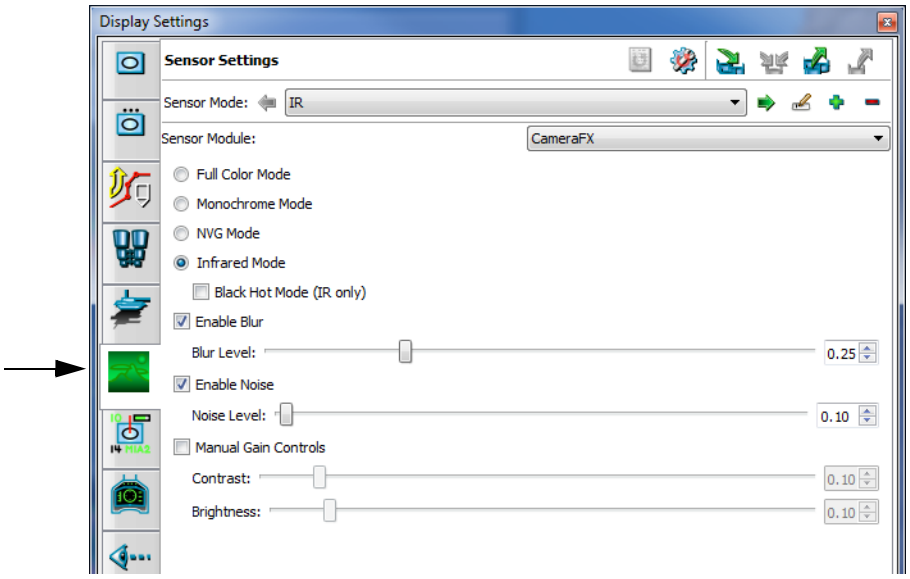


Figure 16-3. Sensor Settings page

3. In the Sensors list, select the sensor that you want to configure.
4. Select the Sensor module from the list.
5. Select the sensor mode. If you select Infrared Mode, select or clear the Black Hot Mode check box.

- It may seem obvious that if, for example, you select NVG from the Sensors list, you would select it for the sensor type. However, VR-Forces lets you add additional sensor configurations to the list and in that case, you would need to specify the correct sensor type for the new sensor. For example, you might create a sensor called NVG-High Contrast. Then you would select NVG as the sensor type and adjust the contrast accordingly.
- Full Color Mode provides natural coloring, as you would see with no sensor modes applied. The benefit of creating a sensor mode using this option is that you can apply noise or blur to the scene.

---

6. Select the check boxes for the effects that you want to be enabled for this sensor configuration.
7. Adjust the values for the enabled effects.

16.4. Adding New Sensors

You can add, rename, and delete sensors.

To add a sensor:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Sensor Settings page (Figure 16-3).
3. On the Sensor line, click the Add button (). A new sensor called sensor_*number* is added. There are no configuration options on the page. (To rename the sensor, click the Rename button () and enter a new name.)
4. In the Sensor Module list, select CameraFX. The available configuration options are added to the page.
5. Select the type of sensor that you want this to be (Full Color, Monochrome, Infrared, or NVG).
6. Set the Blur, Noise, and Gain levels.



17. Lighting Effects

This chapter describes the lighting effects supported by VR-Forces.

Displaying Lighting Effects 17-2
Displaying Dynamic Lighting 17-3
Enabling and Disabling Dynamic Lighting and Shading 17-4
Specifying the Emissive Lighting Scale 17-5
Displaying Ocean Planar Reflections 17-6
Configuring Ocean Planar Reflections 17-7
Lens Flare 17-8
Displaying Crepuscular (Sun) Rays 17-9
Configuring Ambient and Diffuse Lighting 17-10
Configuring the Ambient and Diffuse Scale Values 17-10
Configuring the Material Ambient and Diffuse Values 17-10
Configuring the Skybox Cube Map 17-12
Flashlight Mode 17-12
Displaying Shadows (3D Only) 17-13
Configuring Shadows 17-15
Disabling Lighting and Clouds 17-15

17.1. Displaying Lighting Effects

VR-Forces can simulate a variety of lighting effects, including:

- Dynamic lighting
- Ocean planar reflection
- Lens flare
- Crepuscular (sun) rays.
- Shadows.

Use of lighting effects affects performance. VR-Forces lets you enable and disable some effects so you can adjust the mixture of performance and effects that best meets your needs.

17.2. Displaying Dynamic Lighting

VR-Forces can display light cast by light sources such as headlights, street lamps, and spotlights. Display of dynamic lighting is an observer-specific feature. It is disabled by default.

Dynamic lights are either part of the terrain or part of an entity model. If they are part of an entity model, they can be set permanently on or they can be turned on and off by the appearance bit. VR-Forces entities that have lights can enable them using the Set Appearance set data request.

Figure 17-1 shows a night-time scene with dynamic lighting enabled. The HMMWV's taillights cast light to the rear. The headlights cast light forward. Street lamps and windows are also casting light.



Figure 17-1. Dynamic Lighting enabled



- Light points that are part of the terrain, such as runway lights, automatically turn on at dusk and turn off at dawn.
- Emissive lights, such as automobile headlights, are always on.
- Dynamic lights .....

You cannot change these lighting settings.

To enable dynamic lighting,

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page.
3. In the list of settings, select Dynamic Lights.



You can also enable or disable dynamic lighting by choosing **Observer** → **Dynamic Lights** or clicking the Dynamic Lights button (💡) on the Observer Settings Toolbar.

17.2.1. Enabling and Disabling Dynamic Lighting and Shading

You can change how lighting is rendered by enabling and disabling advanced lighting and shading. Figure 17-2 shows the same scene as the previous section. However in this view dynamic lighting is turned off. Light points are still displayed, but no light is cast on the scene. Turning off dynamic lighting may improve performance for some graphics cards.



Figure 17-2. Dynamic Lighting with advanced lighting disabled

To enable or disable advanced lighting:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Lighting Render Settings page (Figure 17-3).

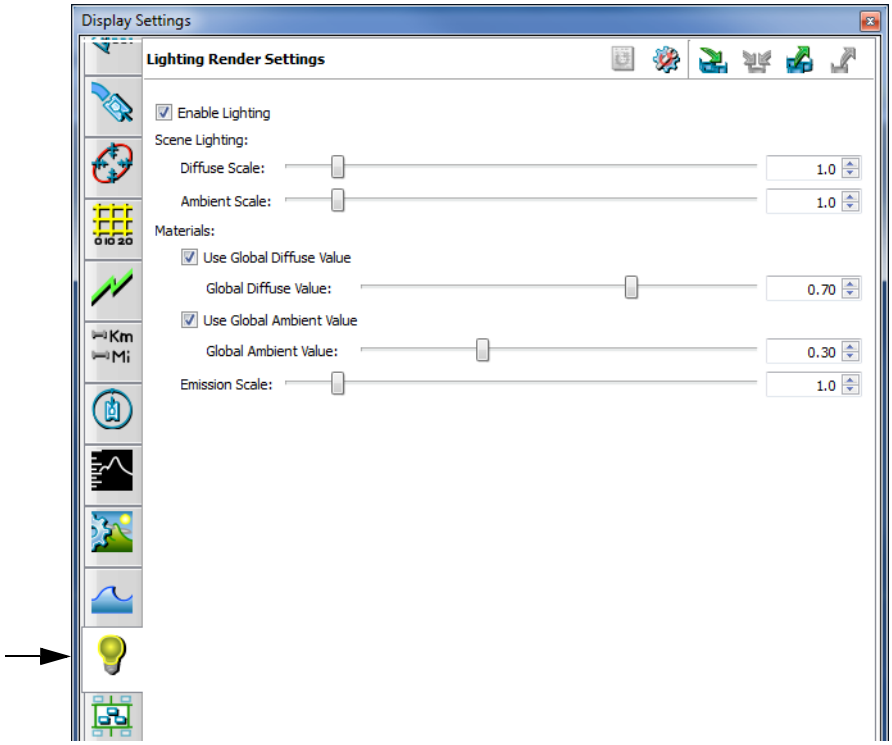


Figure 17-3. Lighting Render Settings

3. Select or clear the Enable Lighting check box.

17.2.2. Specifying the Emissive Lighting Scale

You can change a scale value that controls the brightness of emissive textures. This allows you to make windows, street lights, and other emissive light sources darker or brighter. Increasing the scale value increases the brightness.

To specify the emissive lighting scale value:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Lighting Render Settings page (Figure 17-3).
3. Adjust the Emission Scale slider or change the value in the box.

17.3. Displaying Ocean Planar Reflections

VR-Forces can show the reflection of terrain and objects on water (Figure 17-4). To see planar reflections, dynamic ocean must be enabled. (For details about enabling dynamic ocean, please see Section 15.6.1, "Enabling Marine Effects," in *VR-Forces Scenario Management Guide*.)

Displaying ocean planar reflections degrades performance.



Figure 17-4. Ocean planar reflection

To enable or disable planar reflections:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Ocean Render Settings page (Figure 17-5).

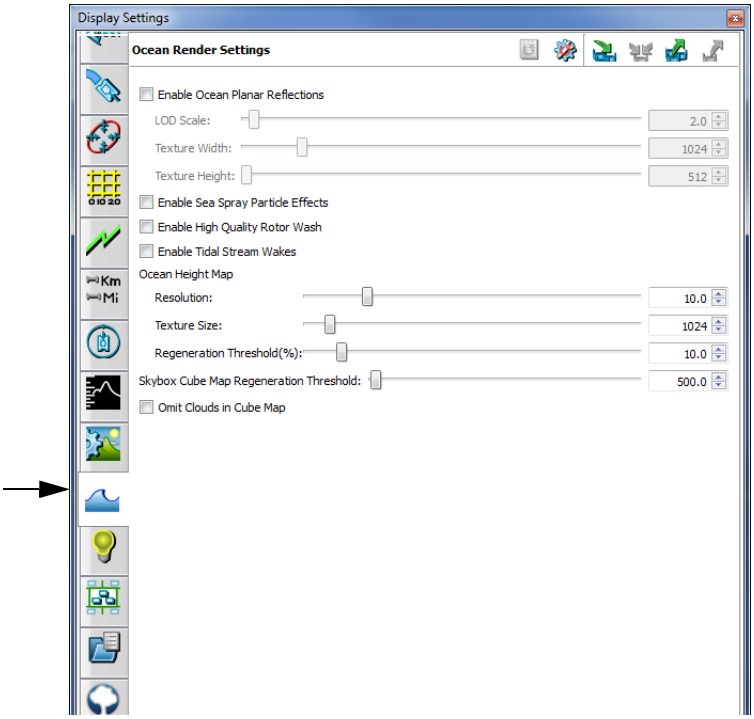


Figure 17-5. Ocean Render Settings

3. Select or clear the Enable Ocean Planar Reflections check box.

17.3.1. Configuring Ocean Planar Reflections

To reduce the performance impact of ocean planar reflections, you can configure the LOD scale. The LOD scale affects the distance from the observer at which VR-Forces switches LODs. The higher the value, the more quickly LODs are switched.

The Texture Height and Texture Width parameters affect the resolution of the ocean planar reflection texture, in pixels. Higher resolution decreases performance.

To configure ocean planar reflections:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Ocean Render Settings page (Figure 17-5).
3. To change the ocean planar reflection LOD scale, adjust the LOD Scale slider under Enable Ocean Planar Reflections, or change the value in the box.
4. To change the width or height of the ocean planar reflection texture, adjust the Texture Width or Texture Height slider under Enable Ocean Planar Reflections, or change the values in the boxes.

17.4. Lens Flare

Lens flare is a visual artifact often seen when looking at light sources such as the sun through a camera lens (Figure 17-6). VR-Forces can simulate lens flare when the observer looks towards the sun.



Figure 17-6. Lens flare

To enable or disable lens flare:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Render Settings page (Figure 17-7).

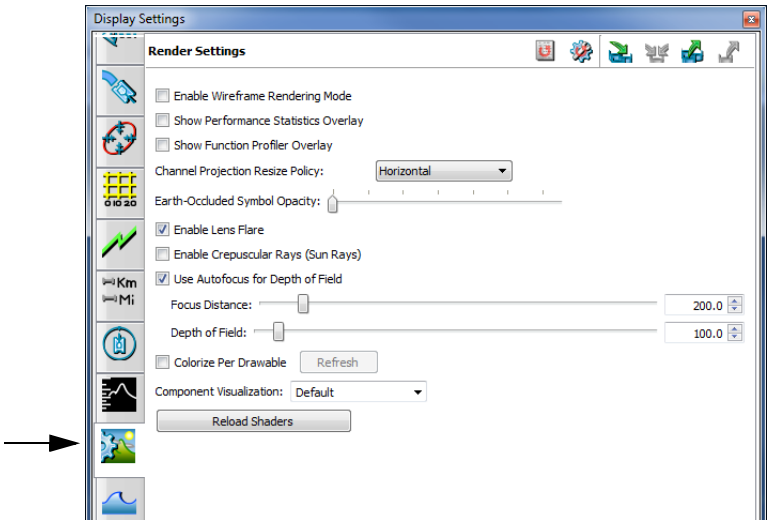


Figure 17-7. Render Settings page

3. Select or clear the Enable Lens Flare check box.

17.5. Displaying Crepuscular (Sun) Rays

VR-Forces can simulate the effect of crepuscular rays (sunbeams) beaming through clouds (Figure 17-8).

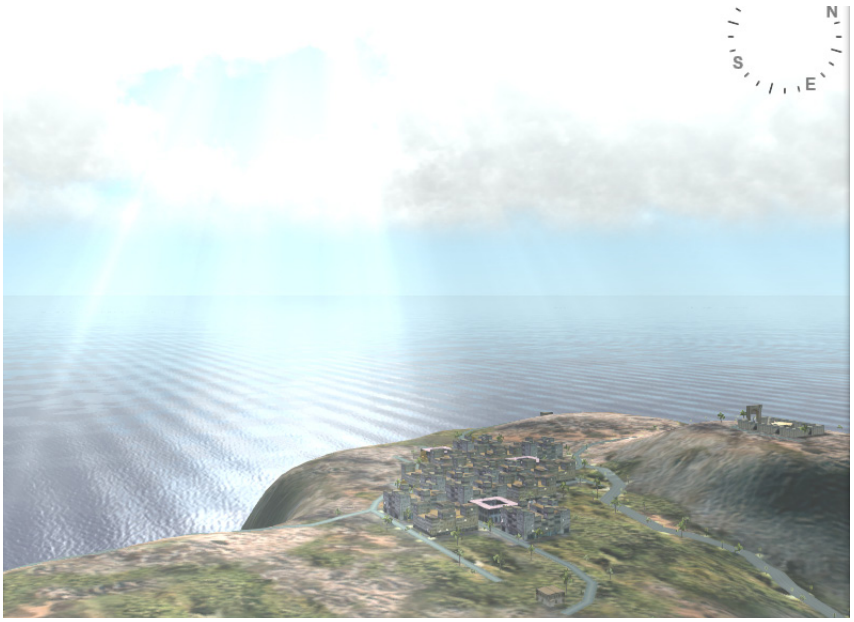


Figure 17-8. Crepuscular rays

To enable or disable crepuscular (sun) rays:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Render Settings page (Figure 17-7).
3. Select or clear the Enable Crepuscular Rays (Sun Rays) check box.

17.6. Configuring Ambient and Diffuse Lighting

Ambient lighting is the lighting available to objects that are not in a direct form of light, such as sunlight. Diffuse lighting is light that is reflected off an object. You can adjust the overall amount of ambient and diffuse lighting in the scene and you can configure whether objects use the material ambient and diffuse values specified in their models or use values provided by VR-Forces.

17.6.1. Configuring the Ambient and Diffuse Scale Values

You can set a scale factor for ambient and diffuse light that is present in the scene. Lower values make objects darker. Higher values make them lighter. The ambient or diffuse scale factor is independent of how material ambient and diffuse values are set.

To specify the ambient and diffuse lighting scale values:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Lighting Render Settings page (Figure 17-9).
3. Adjust the Diffuse Scale slider or change the value in the box.
4. Adjust the Ambient Scale slider or change the value in the box.

17.6.2. Configuring the Material Ambient and Diffuse Values

Polygon materials may specify specular, diffuse, and ambient colors that define the way they respond to incoming light.Polygons in a 3D scene have a material and a texture. The material can have specular, diffuse, and ambient colors defined.

The ambient color specifies the color used when it is in indirect light. Sometimes the ambient values for different objects do not work well together. You can specify that objects use the ambient material value, or let VR-Forces manageglobally specify ambient material values.

In addition to specifying which material ambient and diffuse values to use, you can set a scale factor. Lower values make objects darker. Higher values make them lighter. The ambient or diffuse scale factor is independent of how ambient and diffuse values are set.

To configure the material ambient value:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Lighting Render Settings page (Figure 17-9).

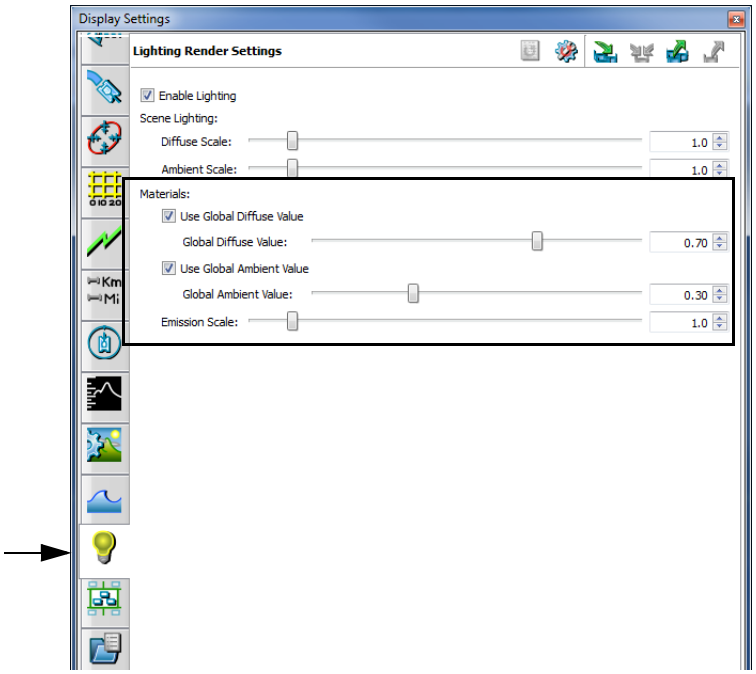


Figure 17-9. Ambient and Diffuse Value settings

3. To use the ambient values specified by objects, Select or clear the Use Material-Global Ambient Value check box. If selected, the ambient value specified by the object is used. If cleared, VR-Forces manages ambient value.To have VR-Forces use a global value, select the Use Global Ambient Value check box.
4. If you selected the Use Global Ambient Value check box, oOptionally, adjust the Global Ambient Scale FactorValue slider or change the value in the box.

To configure the material diffuse value:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Render Settings page (Figure 17-9).
3. Select the Lighting Render Settings page (Figure 17-9).
4. Select or clear the Use Material Diffuse Value check box. If selected, the diffuse value specified by the object is used. If cleared, VR-Forces manages diffuse value.
5. Optionally, adjust the Diffuse Scale Factor slider.
6. To use the diffuse values specified by objects, clear the Use Global Diffuse Value check box. To have VR-Forces use a global value, select the Use Global Diffuse Value check box.

7. If you selected the Use Global Diffuse Value check box, optionally, adjust the Global Diffuse Value slider or change the value in the box.

17.7. Configuring the Skybox Cube Map

A skybox cube map is a graphics technique used to render reflections and sky effects. The skybox cube map is rendered from the location of the observer. As the observer moves, the cube map needs to be recalculated, which is computationally expensive.

VR-Forces lets you configure the distance that the observer has to move (the skybox cube map regeneration threshold) before it recalculates the cube map. Increasing the threshold can improve performance, at the cost of visual quality.

You can also choose to omit clouds in the skybox cube map. Omitting clouds can increase performance.

To configure the skybox cube map:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Ocean Render Settings page (Figure 17-5).
3. Adjust the Skybox Cube Map Regeneration Threshold slider or enter a value in the box.
4. Optionally, select the Omit Clouds in Cube Map check box.

17.8. Flashlight Mode

VR-Forces can light a circular area as if the observer is using a flashlight. You can move the flashlight beam and increase or decrease the diameter of the flashlight beam.

- To turn flashlight mode on or off, press the **`f`** key.
- To move the flashlight beam, hold down the **`Alt`** key and the right mouse button and move the mouse.
- To change the size of the flashlight beam, hold down the **`Alt`** key and move the mouse scroll wheel forward or back.

17.9. Displaying Shadows (3D Only)

VR-Forces can display shadows for entities, props, and terrain features.



Figure 17-10. Shadows

You can enable and disable shadows and you can configure shadow quality.

Enabling shadows can affect performance.

To enable and disable shadows globally:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Shadow Settings page (Figure 17-11).

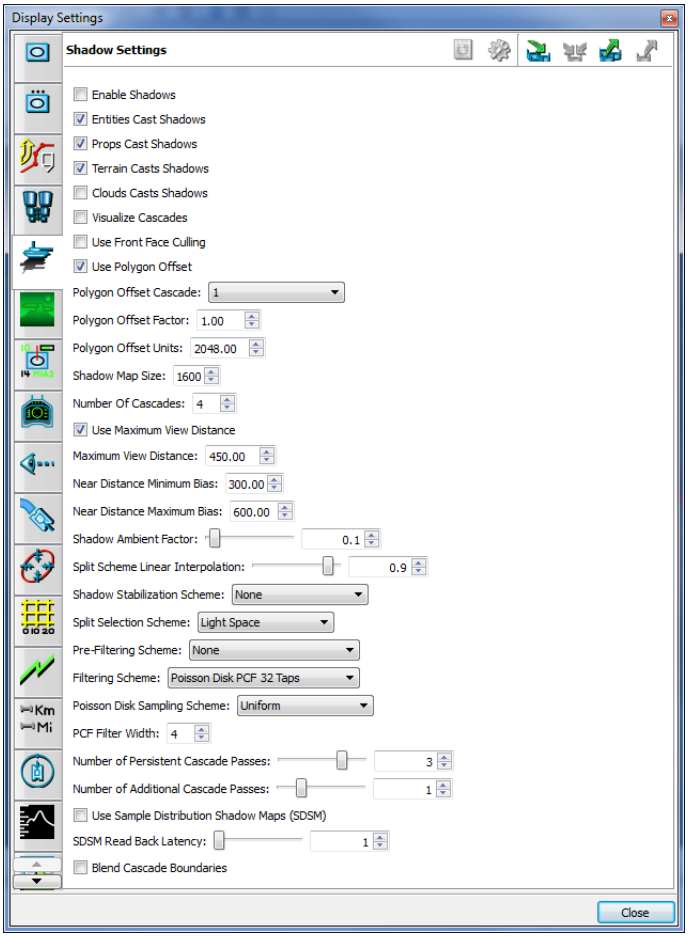


Figure 17-11. Display Settings, Shadow Settings page

3. Select or clear the Enable Shadows check box.



You can also enable and disable shadows by choosing **Settings** → **Shadows** or clicking the Shadows button ( ) on the Display Settings Toolbar.

17.9.1. Configuring Shadows

You can configure which objects cast shadows and parameters that affect performance and the quality of shadows. Shadow parameters include the following:

- Cascades. A scene is split into sections, with those closest to the camera showing the highest resolution shadows. You can configure the number of splits and how the quality of resolution changes as you move away from the camera.
- Polygon offset. These settings shift shadows to minimize Z-fighting.
- Shadow ambient factor. Hides Z-fighting and other visual artifacts.
- Shadow stability. Affects resolution when the camera moves.
- Filtering. These options affect visual quality.
- Sample Distribution Shadow Maps (SDSM). SDSM increases rendering quality at the expense of performance.

To configure how shadows are rendered:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Shadow Settings page (Figure 17-11).
3. To enable or disable shadows for object types, select or clear the Entities Cast Shadows, Props Cast Shadows, or Terrain Casts Shadows check boxes.
4. Change other parameters as desired.

17.10. Disabling Lighting and Clouds

When you are using XR observer mode and you move the observer out from the terrain, the lighting and cloud effects can interfere with your ability to view entities. Therefore, VR-Forces lets you quickly disable lighting effects and clouds. By default, they are enabled in Stealth observer mode and disabled in XR observer mode.

➤ To enable or disable lighting and clouds, choose **Observer** → **Sky**, or click the Show/Hide Sky button () on the Observer Settings Toolbar.



18. Displaying Terrain Information

This chapter describes how to change the display of terrain to provide different types of useful terrain information.

Introduction 18-2
Configuring Visible Surfaces (GDB Soil Types) 18-2
Coloring Terrain Based on Elevation 18-4
    Configuring Elevation Color Settings 18-5
Displaying Grid Lines 18-7
    Configuring Grid Lines 18-8
Displaying Contour Lines 18-9
    Configuring Contour Lines 18-10
Displaying Models and Terrain in Wireframe Mode 18-10
Filtering Earth File Layers 18-12
Displaying Feature Data (2D) 18-13
    Displaying a List of Vector Features 18-14
    Changing the Colors Used for Feature Data 18-15
Displaying Coordinates and Soil Type for a Location 18-16
Viewing the Map Scale 18-17
Terrain Databases Provided with VR-Forces 18-17

18.1. Introduction

As you create and view simulations, you may need information about the terrain. By changing how terrain is displayed, you can get some of this information. If your terrain does not have texture data or an image overlay, features such as elevation coloring can provide a more realistic look to the terrain.

For information about how to compose terrains, please see Chapter 13, *Composing Terrains*, in *VR-Forces Configuration Guide*.

18.2. Configuring Visible Surfaces (GDB Soil Types)

If a terrain patch was loaded from a GDB file, you can display or hide GDB soil types. Figure 18-1 shows the *makland.gdb* terrain file. The view on the left shows all soil types. The view on the right has Building, Grass, Ocean, and Rock surfaces hidden.

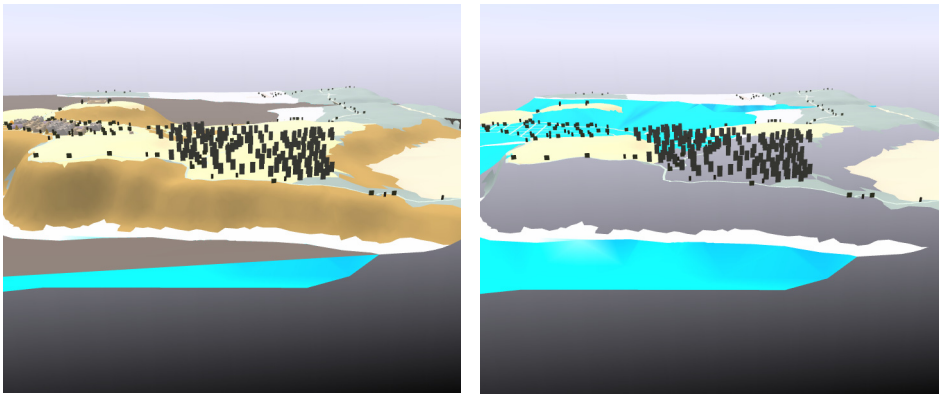


Figure 18-1. Visible surfaces toggled off

To display or hide GDB soil types:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Visible Surfaces page (Figure 18-2).

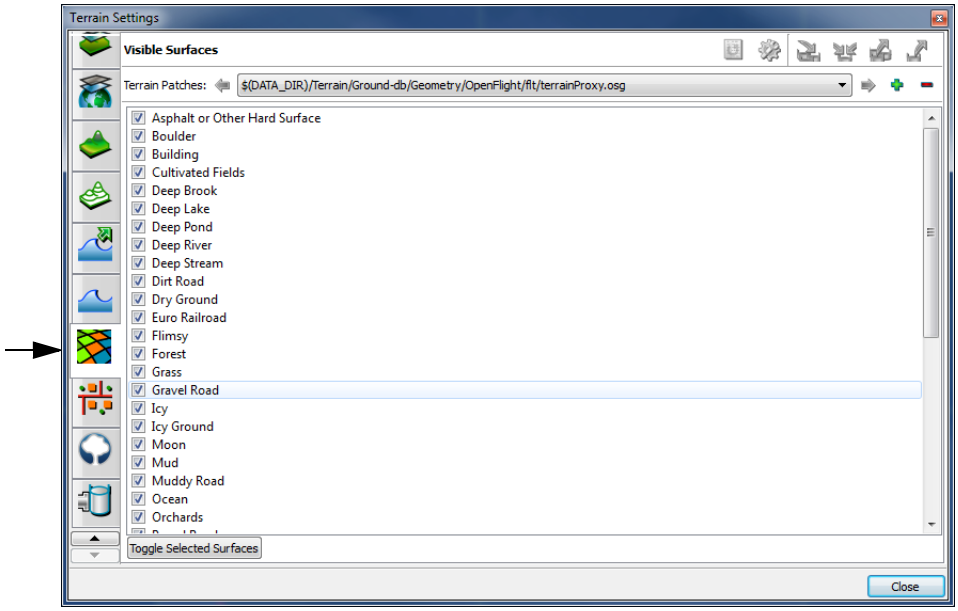


Figure 18-2. Visible Surfaces page

3. In the Terrain Patches list, select the terrain patch that you want to configure.
4. Select the check box for each soil type that you want to be visible.
5. Clear the check box for each soil type that you want to hide.
6. To change the state of multiple soil types at one time:
   a. Select (highlight) the names (not the check boxes) of the soil types whose state you want to change.
   b. Click Toggle Selected Surfaces.

18.3. Coloring Terrain Based on Elevation

You can color the terrain based on elevation.

> To use terrain coloring, you must disable advanced lighting and shading (on the Render Settings page).

To enable or disable coloring of terrain by elevation:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Elevation Color page (Figure 18-3).

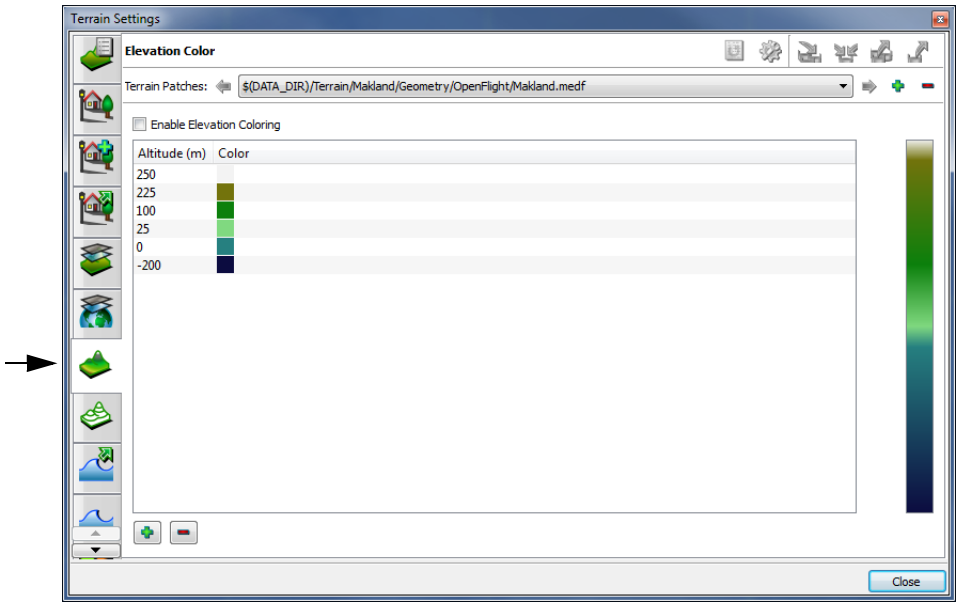


Figure 18-3. Elevation Color page

3. In the Terrain Patches list, select the terrain patch for which you want to enable elevation coloring.
4. Select or clear the Enable Elevation Coloring check box.

18.3.1. Configuring Elevation Color Settings

You can specify the color applied to terrain at a given altitude. You can add altitude points and remove them.

Inserting an Altitude Point

To insert an altitude point:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Elevation Color page (Figure 18-3).
3. In the Terrain Patches list, select the terrain patch for which you want to insert an altitude point.
4. Select an altitude point in the list that is above or below the altitude point you want to add.
5. Click Insert Point. A new point is added with the same altitude as the one you selected.
6. Double-click the new altitude point to make it editable.
7. Set the altitude point to the altitude that you want.
8. Click away from the altitude point to set it.

Deleting an Altitude Point

To delete an altitude point:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Elevation Color page (Figure 18-3).
3. In the Terrain Patches list, select the terrain patch for which you want to delete an altitude point.
4. Select the altitude point that you want to delete.
5. Click Delete Point.

Changing the Color of an Altitude Point

To change the color of an altitude point:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Elevation Color page (Figure 18-3).
3. In the Terrain Patches list, select the terrain patch for which you want to change the color of an altitude point.
4. Double-click the color swatch for the altitude point whose color you want to change. A color chooser dialog box opens.
5. Select or specify a new color.
6. Click OK.

18.4. Displaying Grid Lines

VR-Forces can display grid lines in the 2D view. Grid-lines are an observer-specific setting.

To enable or disable grid lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Observer Settings page (Figure 18-4).

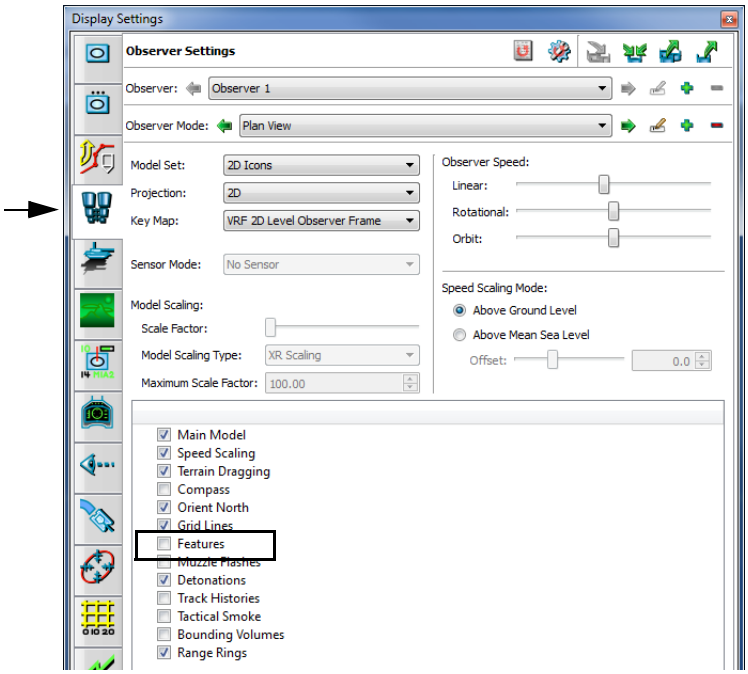


Figure 18-4. Observer Settings page

3. Select Plan View mode (or a locally created 2D observer mode).
4. Select or clear the Grid Lines check box.



You can also enable or disable grid lines by choosing **Observer** → **Grid Lines** or clicking the Grid Lines button () on the Observer Settings Toolbar.

18.4.1. Configuring Grid Lines

You can configure the coordinate system used for grid lines, the precision, and the spacing. Spacing can be fixed, in meters, or determined by VR-Forces using a specified number of screen pixels. Fixed spacing can result in a very tight grid and unreadable labels as you zoom out. Auto spacing does not have this problem. However, the grid lines will constantly recalculate as you zoom in and out.

To configure grid lines:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Grid Lines Settings page (Figure 18-5).

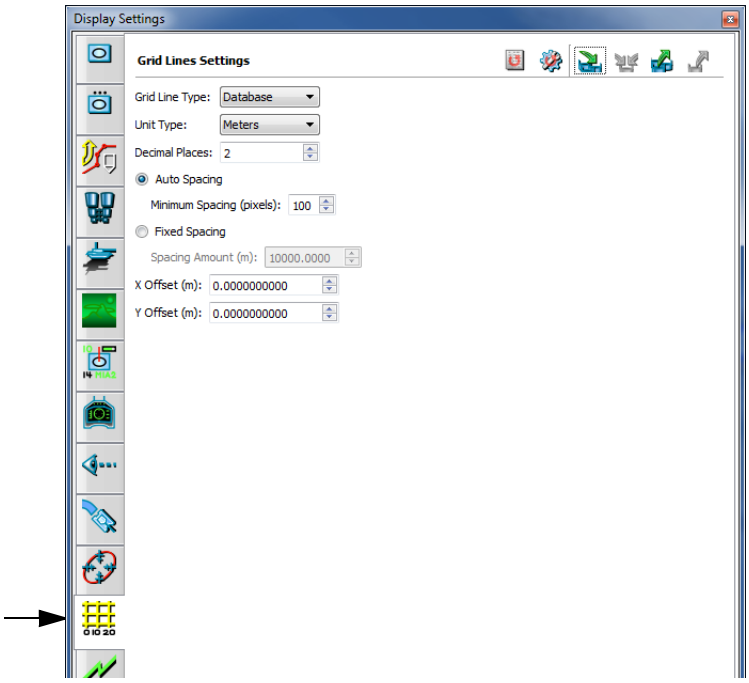


Figure 18-5. Grid Lines Settings page

3. To change the coordinate system used for grid lines, select an option in the Grid Line Type list. The list displays grid line types that are appropriate for the type of database loaded. (If no database is loaded, the coordinate system is set to Flat Earth.)
4. To specify the unit type for the grid line labels, select an option on the Unit Type list.
5. To specify the precision of the grid line labels, type a value in the Decimal Places box.
6. To change the spacing type, select AutoSpacing or Fixed Spacing and specify a value for the chosen spacing type.

7. To specify an offset from the origin, change the values for X Offset and Y Offset.

18.5. Displaying Contour Lines

You can display contour lines and intermediate contour lines on the terrain. Intermediate contour lines divide the contour line interval into fifths (Figure 18-6). You can configure the thickness of contour lines and the interval between them.



Figure 18-6. Contour lines and intermediate contour lines

To display contour lines:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Contour Lines page (Figure 18-7).

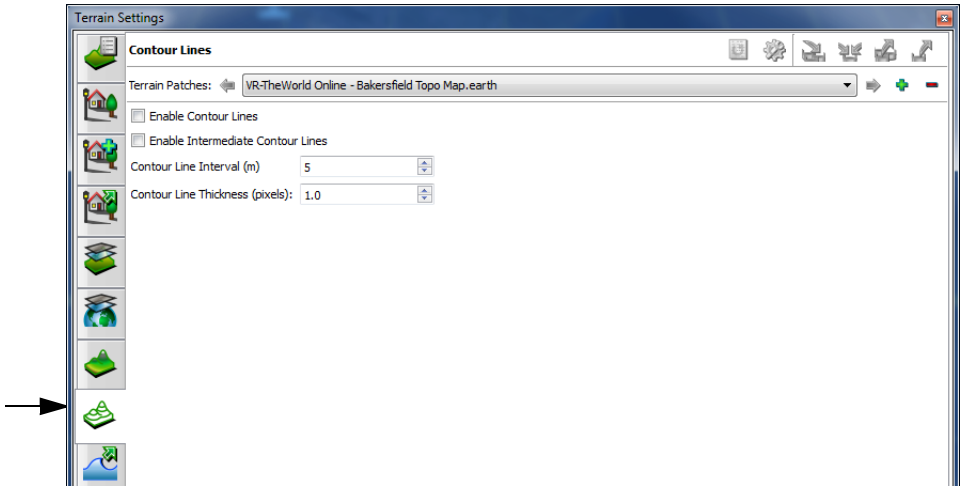


Figure 18-7. Contour Lines page

3. In the Terrain Patches list, select the terrain patch for which you want to enable or disable the display of contour lines.
4. Select Enable Contour Lines.
5. To enable intermediate contour lines, select Enable Intermediate Contour Lines.

18.5.1. Configuring Contour Lines

You can configure the thickness of contour lines and the interval between them.

To configure contour lines:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Contour Lines page (Figure 18-7).
3. In the Terrain Patches list, select the terrain patch for which you want to configure the thickness and interval of contour lines.
4. In the Contour Line Interval text box, specify the distance, in meters, between contour lines.
5. In the Contour Line Thickness text box, specify the thickness, in pixels, for contour lines.

18.6. Displaying Models and Terrain in Wireframe Mode

In wireframe mode, VR-Forces displays the polygons that make up the terrain and objects (Figure 18-8).



Figure 18-8. Wireframe off and on

To enable or disable wireframe mode:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Render Settings page (Figure 18-9).

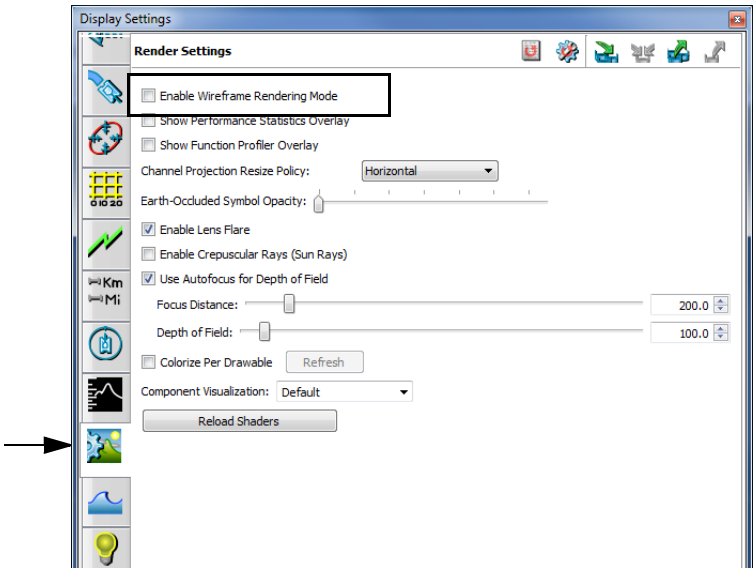


Figure 18-9. Render Settings page

3. Select the Enable Wireframe Rendering Mode check box.

You can also enable and disable wireframe mode by choosing **Settings** → **Wireframe Rendering** or clicking the Wireframe Rendering button ( ) on the Display Settings Toolbar.

18.7. Filtering Earth File Layers

When you load terrain using an Earth file, imagery and feature data is specified in image elements and model elements. You can load multiple instances of these types of data, resulting in a rich, complex terrain. However, if you are only interested in certain types of terrain features, such as roads or rivers, displaying all of the terrain data can obscure the data that you are most interested in. To help declutter the map, you can filter out image and model layers globally or by observer mode.

- Filtering an earth file layer affects all open windows. You cannot hide a layer in one window and leave it open in a different one.
- Filtering only affects the front-end. All layers are loaded in the back-end.

To filter out earth file layers:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Earth Layers page (Figure 18-10).

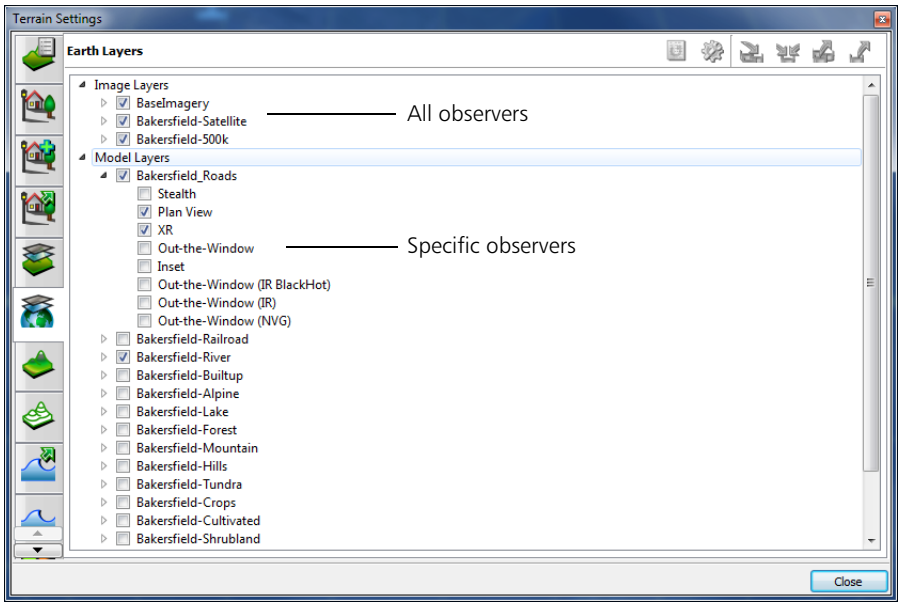


Figure 18-10. Earth Layers page

3. To display or hide an image or model layer for all observer modes, select or clear its check box.
4. To display or hide an image or model layer for a particular observer mode, expand the entry for the layer you want to change and select or clear the check box for the observer mode you want to change.

18.8. Displaying Feature Data (2D)

If a terrain contains feature data, you can display it in the 2D view. Figure 18-11 shows a section of the Little Pond terrain with features disabled and enabled.



Figure 18-11. Feature data

This setting only displays features imported from GDB files or shapefiles. It does not display features streamed from a terrain server. If you load features using an earth file, they are drawn (or not drawn) based on the configuration information in the file.



If your terrain has many features, displaying them can take a long time.

➤ To display feature data, choose **Observer** → **Features**, or click the Feature Data button () on the Observer Settings Toolbar.

18.8.1. Displaying a List of Vector Features

If a terrain has vector features, they are listed on the Vector Network tab of the Terrain Panel. If you select a feature in the list, it is highlighted in the display, and vice versa (Figure 18-12).

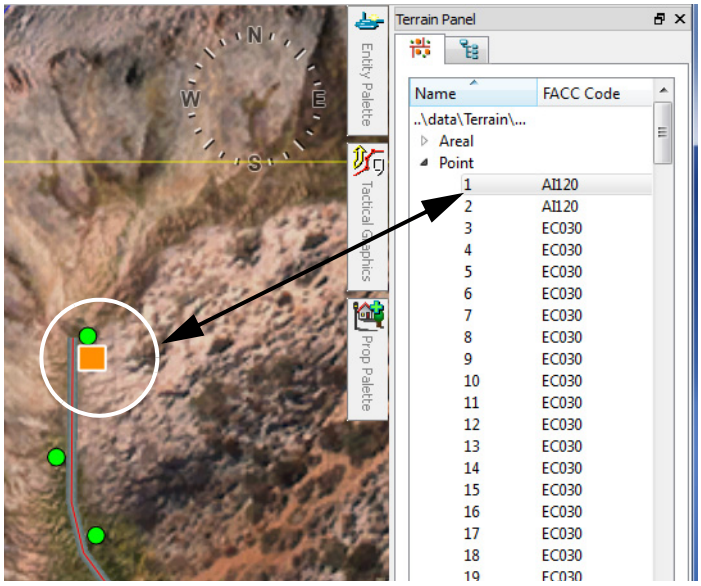


Figure 18-12. Vector feature selected

To display the feature list:

1. Choose **View** → **Terrain Panel**. The Terrain Panel opens.
2. Select the Vector Network tab (Figure 18-12).
3. Expand the top-level categories to see the lists of individual features.

18.8.2. Changing the Colors Used for Feature Data

If a terrain database contains feature data, you can change the colors used to display it. In Figure 18-13, the Woodland feature in the Little Pond terrain was changed to pink. Compare it to the default color (Figure 18-11).



Figure 18-13. Feature data with color change

To change the color of feature data:

1. Choose **Settings** → **Terrain**. The Terrain Settings dialog box opens.
2. Select the Feature Settings page (Figure 18-14).

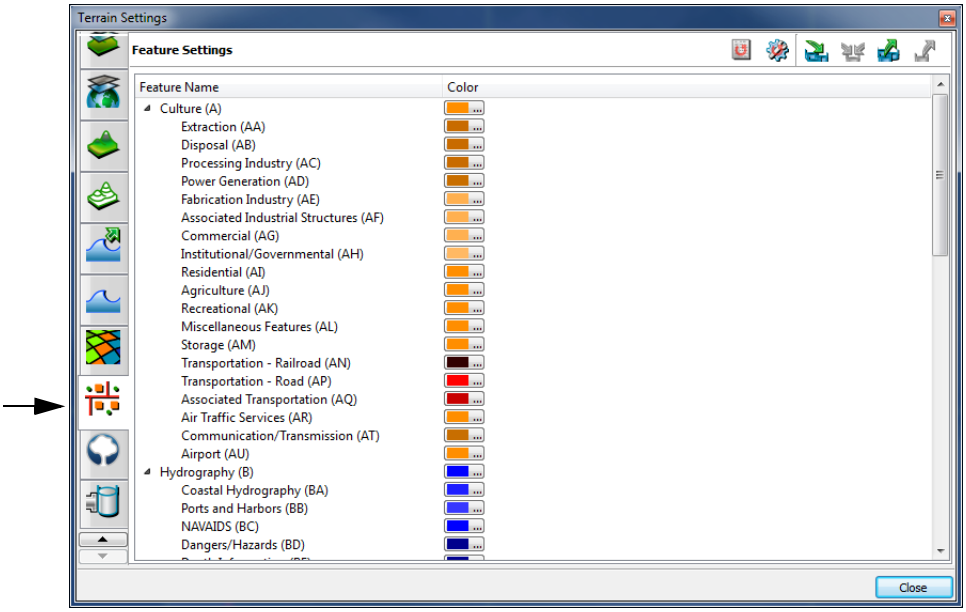


Figure 18-14. Feature Settings page

3. Click the color swatch for the feature that you want to change. A color picker dialog box opens.
4. Select the color that you want to use.
5. Click OK.

18.9. Displaying Coordinates and Soil Type for a Location

You can display the coordinates and soil type (if defined) of features and any point in the terrain. Soil type must be provided by the simulation engine. If you do not query the simulation engine, the Last Clicked Location Panel just displays coordinates using data from the front-end.

The Last Clicked Location Panel becomes unavailable if you query the simulation engine and it takes more than 1/2 second for a response.

To check the coordinates and soil type of a feature or location:

1. If the Last Clicked Location Panel is not open, choose **View** → **Last Clicked Location**.

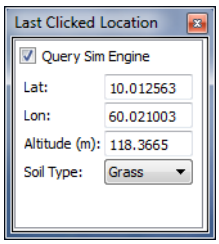


Figure 18-15. Last Clicked Location Panel

2. If you want soil type information, select Query Sim Engine.
3. Click a point on the terrain or click a feature. Information is displayed in the Last Clicked Location Panel.

18.10. Viewing the Map Scale

If you want to know the relationship between distance on your computer screen and distance on the terrain, you can view the Map Scale Toolbar (Figure 18-16). It shows the relationship of one inch of screen to distance, in the currently selected distance unit, on the terrain. The scale changes as you zoom in and out.

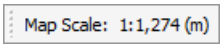


Figure 18-16. Map Scale Toolbar

➤ To enable or disable display of the Map Scale Toolbar, choose **View** → **Map Scale Toolbar**.

18.11. Terrain Databases Provided with VR-Forces

Table 18-1 describes the terrain databases provided with VR-Forces. Some have variants, which indicate the coordinate system or other variations. VR-Forces also has a supplemental data package that includes additional terrains. Please contact your MÄK salesperson for download details.

Table 18-1: Terrain databases shipped with VR-Forces

Database	Variation	Description
Ala Moana	Ala Moana.mtf	A high-resolution area of Honolulu, Hawaii, USA, set into a raster map of the world. This terrain is stored locally and does not require an internet connection.
Berkeley	Berkeley.mtf	UTM. OpenFlight. Feature data. Shows the area around Berkeley, California, USA. It is provided courtesy of Terrex.
Ground	Ground-db.mtf	UTM. OpenFlight. A generic relatively flat ground terrain.
HarrisAfghan-Village	HarrisAfghanVillage-FlatEarthMetaFlight.mtf	Paged terrain using flat earth projection. A terrain of Afghanistan with a high resolution inset of a small village. This terrain was created in VR-Vantage from source data. The terrain data is provided courtesy of Harris Inc.
	HarrisAfghanVillage-FlatEarth.mtf	A version of this terrain based on a GDB file.
	HarrisAfghanVil-lageGdal.mtf	A version of the HarrisAfghanVillage terrain inset into VR-TheWorld and paged in from local source data.
	HarrisAfghanVillageGeo-centricMetaFlight.mtf	Paged terrain using geocentric coordinates.

Table 18-1: Terrain databases shipped with VR-Forces

Database	Variation	Description
Kabul	Kabul.mtf	UTM. OpenFlight. The area around Kabul, Afghanistan.
Little Pond	LittlePond.mtf	Flat earth. DTED. Shape data. Imagery. A detailed suburban inset in a larger, less detailed terrain covering Boston, Massachusetts, U.S.A. and points west. This terrain was created in VR-Vantage using source data as described in "Composing a Terrain through the GUI," on page B-2, in *VR-Forces Configuration Guide*.
Local World Procedural Terrain	Local World Procedural Terrain.mtf	A database that generates terrain procedurally based on soil types. It does not use a satellite image.
Makland	Makland.mtf	UTM. OpenFlight. A fictional terrain created by MÄK. It includes a variety of terrain and feature data, including underwater features. It is placed in the Indian Ocean.
	MaklandCgfTopoMap.mtf	The *makland.gdb* terrain with a topographic map overlay.
	MaklandLocalWorld.mtf	Geocentric. Low resolution local terrain.
Nellis	Nellis.mtf	UTM. An example of a SAF terrain suitable for simulation, but not typically used in 3D imaging. A terrain covering the area around the Nellis Air Force Base, in Nevada, U.S.A.
SanLuisObispo	SanLuisObispo.mtf	Flat earth. DTED. Imagery. A terrain that covers a portion of California, U.S.A.
	SanLuisObispo-TopoGdal.mtf	Geocentric. San Luis Obispo in topographic coordinates as local streaming terrain using a *.earth* file.
TerraSim_SampleUrban	TerraSim_SampleUrban.mtf	UTM. An urban database based on Pittsburgh, Pennsylvania, U.S.A., with a highly detailed segment that includes feature data and buildings with multiple floors. It is provided courtesy of TerraSim.
	TerraSim_SampleUrban-MetaFlight.mtf	A MetaFlight version of this terrain.

Table 18-1: Terrain databases shipped with VR-Forces

Database	Variation	Description
VR-TheWorldOnline	VR-TheWorld Online.mtf	Lets you quickly connect to the publicly available VR-TheWorld Server for streaming terrain.
	VR-TheWorld Online - Little Pond.mtf	VR-TheWorldOnline with the Little Pond terrain inset.
	VR-TheWorld Online - MAK Earth - Base.mtf	VR-TheWorldOnline with high resolution data for Hawaii, Boston, and other areas, but does not support streaming features and some other capabilities.
	VR-TheWorldOnline - MAK Earth.mtf	VR-TheWorldOnline with support for high resolution areas and streaming features. This is the most performant version of VR-TheWorldOnline. This terrain is not supported on 32-bit versions of VR-Forces.
	VR-TheWorldOnline - Massachusetts S-57.mtf	VR-TheWorldOnline with S-57 data (as shapefiles) for Boston harbor. It also demonstrates streaming buoys and beacons.
	VR-TheWorldOnline - VR-Village Afghanistan.mtf	VR-TheWorldOnline with the VR-Village terrain inset in Afghanistan.
	VR-TheWorldOnline - Bakersfield Topo Map.mtf	VR-TheWorldOnline with topographical maps for the area around Bakersfield, California, USA. This terrain contains some feature data that entities in aggregate-level scenarios can take advantage of.
VR-Village	VR-Village.mtf	GDB, feature data, OpenFlight terrain. *VR-Village* is a fictional database created by VT MÄK. It has a highly detailed small town in a larger, sparsely detailed desert environment. The town includes multi-level buildings, underground tunnels, and a road that tunnels through a mountain. The coordinates place it off the west coast of Africa.
	VR-VillageNoSpeedTrees.mtf	VR-Village with no SpeedTrees for improved performance.
World	WorldFlatEarthNoElevation.mtf	Flat earth. Low resolution projected image of the world.
	WorldGeocentric.mtf	Geocentric. Low resolution projection of the world.



19. Remote Graphics

Remote graphics are graphics that are drawn onto the scene by remote programs that use the VR-Vantage Remote Draw API. This chapter explains how to display them in VR-Forces.

Displaying Remote Graphics 19-2
Viewing a List of Remote Graphics 19-4

19.1. Displaying Remote Graphics

VR-Forces can display graphics sent from other applications using the VR-Vantage Remote Draw API (Figure 19-1).

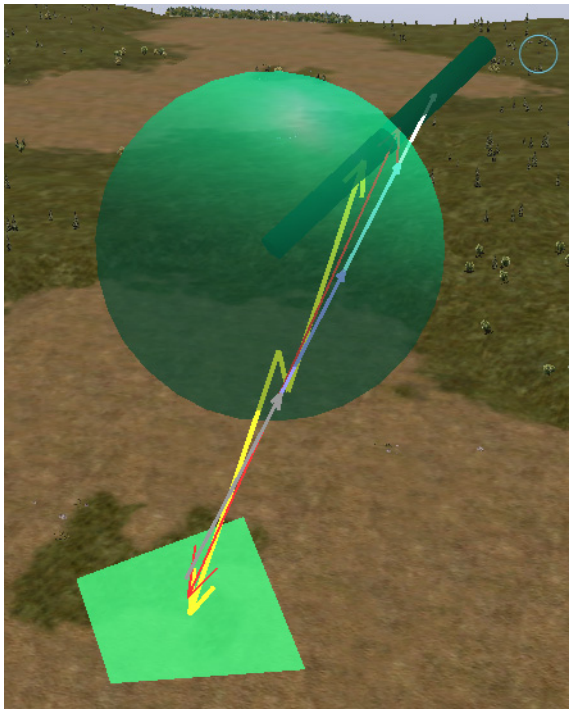


Figure 19-1. Remote graphics

To enable or disable the display of remote graphics:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Tactical Graphics Display Settings page (Figure 19-2).

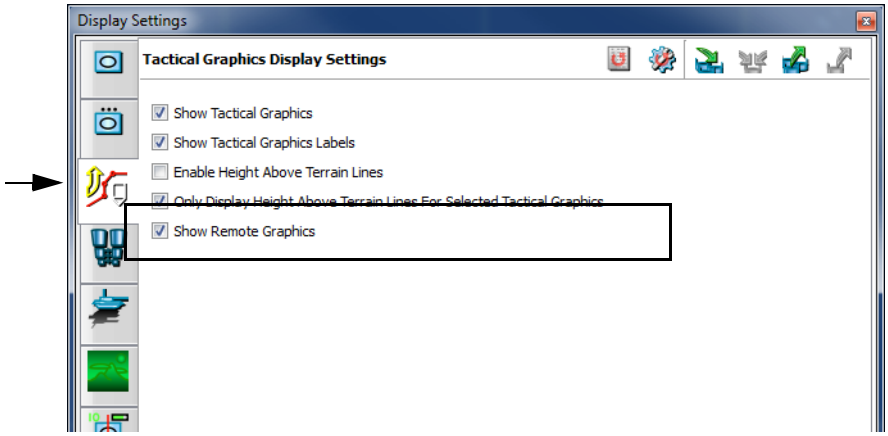


Figure 19-2. Tactical Graphics Display Settings page

3. Select or clear the Show Remote Graphics check box.



You can also enable and disable remote graphics by choosing **Settings** → **Remote Graphics** or by clicking the Remote Graphics button () on the Display Settings Toolbar.

19.1.1. Viewing a List of Remote Graphics

Remote graphics can be organized into sets. There is no fixed organization. Labeling of sets is entirely at the discretion of the programmer who develops the application that generates the graphics. The Remote Graphics Panel displays remote graphics in a hierarchy by set.

The VR-Vantage Remote Draw API lets developers send data about objects as attribute:value pairs, for example Speed:30 KM. The attributes are organized by type using attribute group templates. You can enable and disable the display of attribute information by group and by individual attribute. Figure 19-3 illustrates the attributes for the Entity and Packet attribute group templates displayed in Figure 19-4.

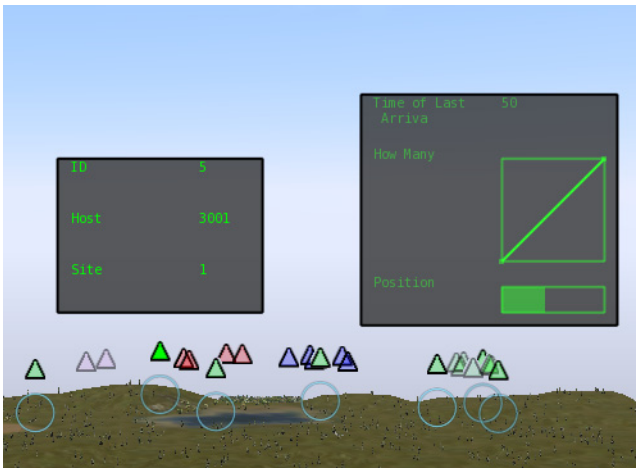


Figure 19-3. Remote graphics attribute group display

To view a list of remote graphics:

1. Choose **View** → **Remote Graphics Panel**. The Remote Graphics Panel is displayed (Figure 19-4). The Remote Graphics Sets tab lists remote graphics by set.

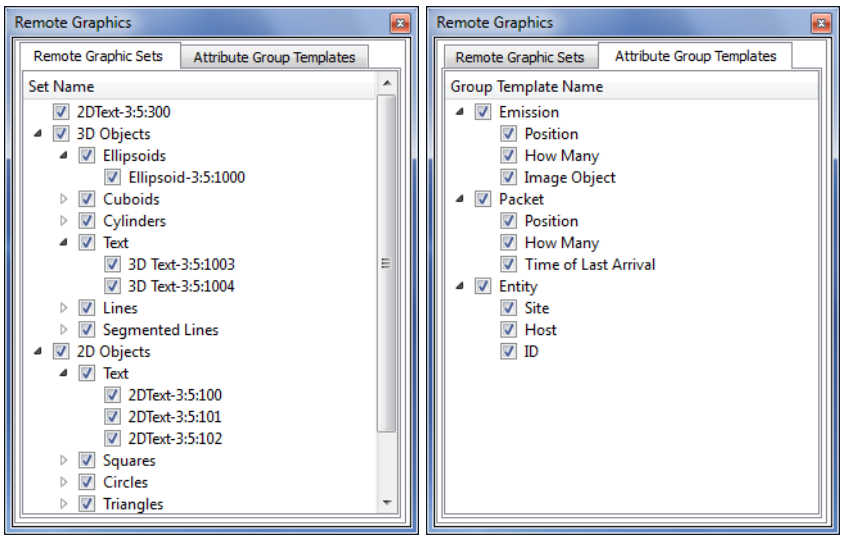


Figure 19-4. Remote Graphics Panel

2. To hide or display a remote graphic set or individual graphic, check or clear its check box in the Remote Graphics Sets list.
3. To view Attribute Group Templates, select the Attribute Group Templates tab. To see the attribute display, mouse over an object for which display data is available.



A. Common GUI Controls and Behaviors

This appendix provides details about using the various panels and toolbars in the front-end.

Dockable Panels and Toolbars A-2
Docking and Undocking a Panel A-2
Docking and Undocking a Toolbar A-3
Displaying and Hiding Panels and Toolbars A-3
Restoring the Layout of Toolbars and Panels A-4
Using Tear-Off Menus A-5
Viewing the Window in Full Screen Mode A-6
Using Context-Sensitive Menus A-6
Common VR-Forces Buttons A-7
Adding Text to the Title Bar A-8
Specifying Display Units A-9

A.1. Dockable Panels and Toolbars

VR-Forces has floating, dockable panels for many of its important functions (Figure A-1). It also has toolbars that let you quickly access the commands available on the menus.

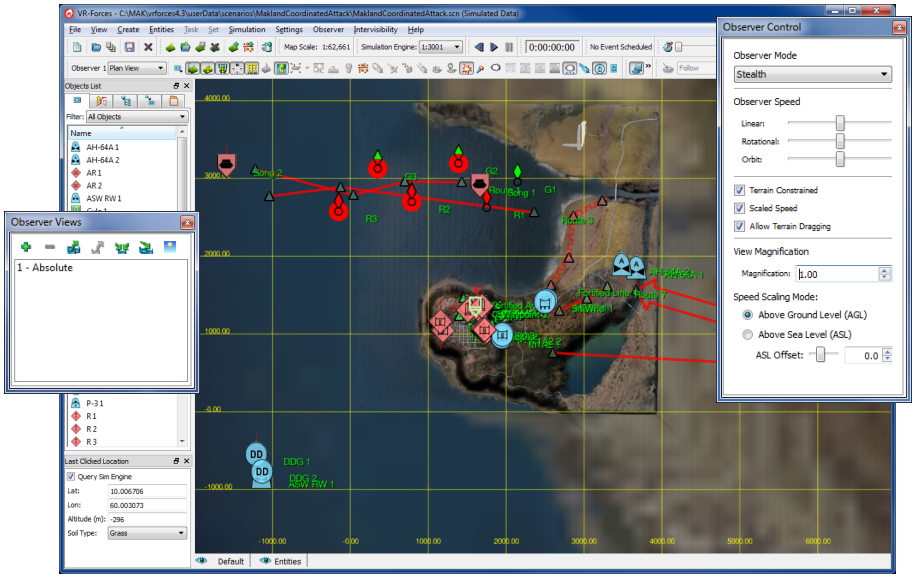


Figure A-1. VR-Forces window and undocked panels

A.1.1. Docking and Undocking a Panel

To undock a panel:

1. Place the cursor over the panel's title.
2. Press and hold the left mouse button.
3. Drag the panel away from the window.

➤ To dock a floating panel, double-click its title bar.

A.1.2. Docking and Undocking a Toolbar

To undock a toolbar:

1. Place the cursor over the handle.



Figure A-2. Toolbar handle

2. Press and hold the left mouse button.
3. Drag the toolbar away from the window.

➤ To dock a toolbar, drag it back to the toolbar strip. The other toolbars will move to create a space to drop in the toolbar (Figure A-3).

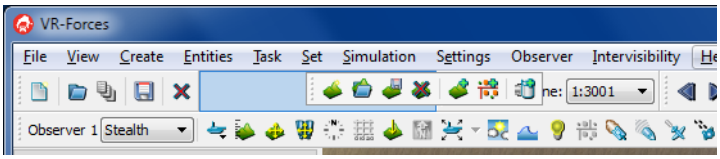


Figure A-3. Docking a toolbar

A.1.3. Displaying and Hiding Panels and Toolbars

You can hide any of the panels or toolbars.

➤ To hide or view a panel or toolbar, select it on the View menu.

A.1.4. Restoring the Layout of Toolbars and Panels

VR-Forces saves the layout of toolbars and panels when you shut down. (It does not save the size and location of the main window.) You can restore the factory layout or the last saved layout.

To restore the layout of toolbars and panels:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the GUI Layout page (Figure A-4).

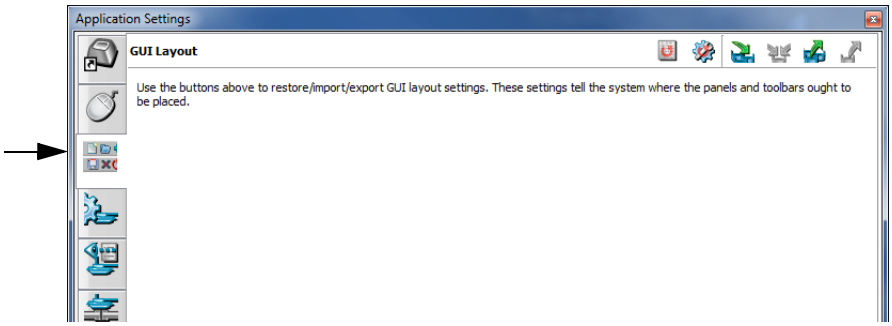


Figure A-4. GUI Layout page

3. Click the settings button for the operation you want to perform. (For a summary of common settings buttons, please see "Managing VR-Forces Settings," on page 5-22.)

A.2. Using Tear-Off Menus

You can "tear-off" the Create menu and leave it displayed on the desktop.

To tear off the Create menu:

1. Click the Create menu.
2. Move the cursor over the grab bar (dashed line) at the top of the menu to highlight it (Figure A-5).
3. Right-click. The menu is displayed as a window that you can leave open and reposition.

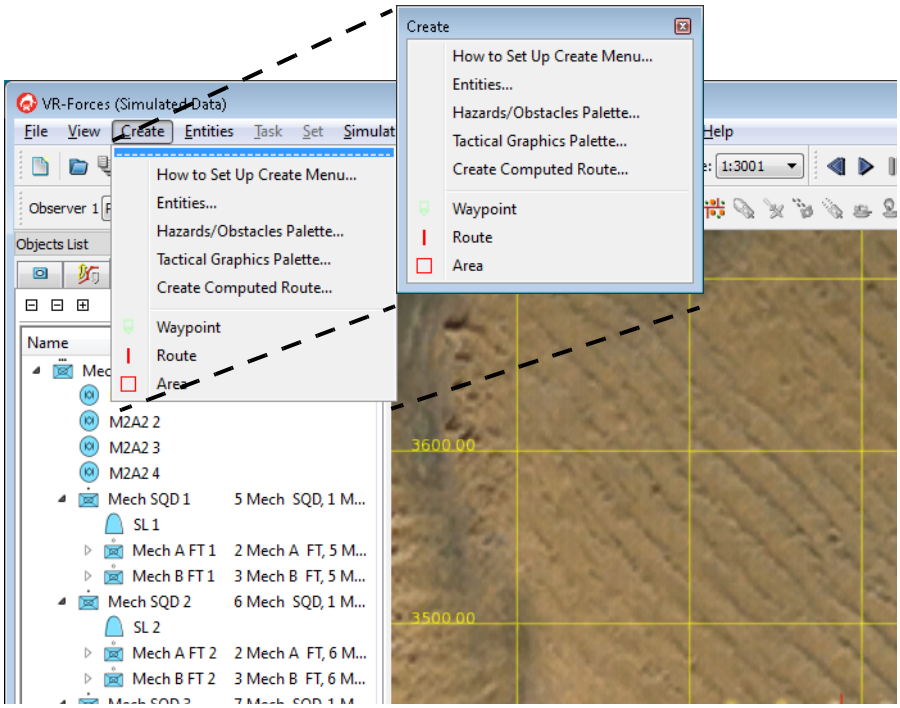


Figure A-5. Tearing off a menu

A.3. Viewing the Window in Full Screen Mode

You can display VR-Forces so that it covers the entire screen and does not show the title bar or border. This mode is intended for use in demonstrations. When menus and panels are hidden, you can still use all keyboard controls, and you can still click entities in the window to attach the observer to them.

You can configure VR-Forces to start in full screen mode with the `--fullScreen` command-line option.

➤ To view VR-Forces in full screen mode, choose **View** → **Show Full Screen**, or press **`Ctrl+Enter`**.

➤ To exit full screen mode, press **`Ctrl+Enter`**.

A.4. Using Context-Sensitive Menus

VR-Forces has context sensitive menus (also called popup menus or right-click menus) that you can display by clicking the right mouse button.

If you right-click an entity, tactical graphic, or other scene object, the menu displays the actions that you can take.

If you right-click the menu bar, you can select the commands available on the View menu.

A.5. Common VR-Forces Buttons

VR-Forces dialog boxes have a common set of buttons. The callouts in Figure A-6 correspond to the numbers in the list of button descriptions.



Figure A-6. Common buttons

- Add an item (definition, model, and so on) (1).
- Remove the selected item (2).
- Copy the selected item and add the copied version to the list (3).
- Select a filter of some type (4).
- Rename the selected item (5).
- Add from a list (6).
- Move up in the list (7).
- Move down in the list (8).
- Select the previous item (9).
- Select the next item (10).
- Add an attribute (11).
- Remove local attribute definition and use the parent's definition (12).
- Search (13).

A.6. Adding Text to the Title Bar

VR-Forces displays the name of the scenario in the title bar. You can display additional text of your choice if you want to. The text is independent of the scenario. It is an application-wide setting.

To add text to the title bar:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the Session Options page (Figure A-7).

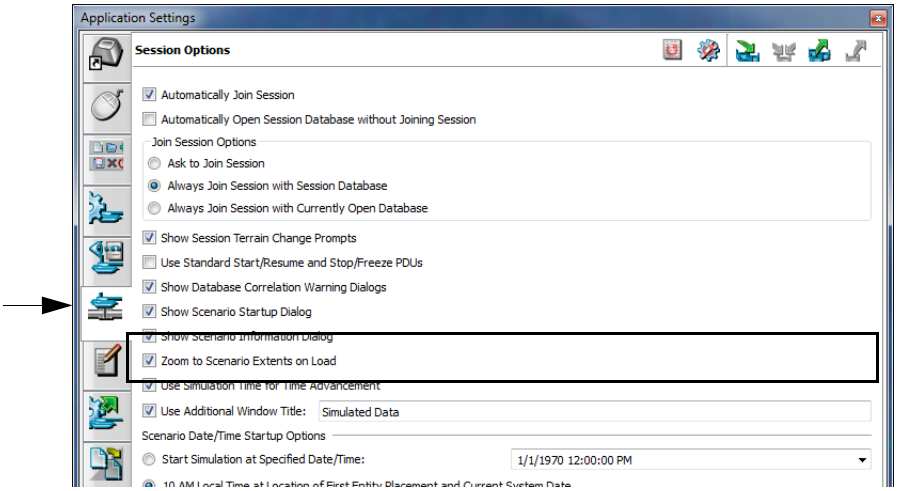


Figure A-7. Session Options page

3. Select the Use Additional Window title check box.
4. Type the text you want to display in the text box.

A.7. Specifying Display Units

You can specify the units used to display coordinate system, distance, altitude, and other measurements. Display units are configured by measurement type. This means that you can use different measurement units for different types of data. You can also specify the number of decimal places for measurement values. When you change a measurement display unit, the changes take effect immediately. They are saved and applied the next time you run VR-Forces.



The following coordinate system and unit display information is independent of (and therefore not affected by) the settings described in this section:

- The coordinate system information for the observer location in the Observer Information Panel.
- Grid line coordinates, units, and precision.

The types of data and the display objects that they affect are as follows:

- Coordinate system
- Distance
- Altitude
- Velocity (speed)
- Acceleration
- Rotational velocity
- Orientation (heading).

Table A-1 describes the coordinate systems that you can specify.

Table A-1: Coordinate systems supported

Coordinate system	Description
Latitude/Longitude (degrees:minutes:seconds)	Location is displayed as two position fields and one altitude field. Displays coordinates in the latitude and longitude using the geodetic WGS84 coordinate system. Each angle will be displayed in degrees:minutes:seconds with seconds displaying base 10 fractional seconds.
Latitude/Longitude (decimal radians	Location is displayed as two position fields and one altitude field. Position coordinates in the latitude and longitude using the geodetic WGS84 coordinate system. Each angle will be displayed in decimal radians.
Geocentric	Location is displayed as three position fields. Position coordinates are displayed in the geocentric coordinate system. The position fields are always in meters.

Table A-1: Coordinate systems supported

Coordinate system	Description
UTM	Location is displayed as two position fields and one altitude field. Position coordinates are displayed in the Universal Transverse Mercator system. The first position field displays the zone and the x location in meters. The second position field displays the y position in meters.
MGRS	Location is displayed as two position fields and one altitude field. Position coordinates are displayed in the Military Grid Reference System. The first position field displays the zone. The second position field displays the grid location. The precision controls the number of digits used in the grid display.
Database	Location is displayed as three position fields. Location is displayed using VR-Forces's current internal cartesian database system. The position fields will be displayed using the current distance units.

To specify measurement units:

1. Choose **Settings** → **Display**. The Display Settings dialog box opens.
2. Select the Display Units Settings page (Figure A-8).

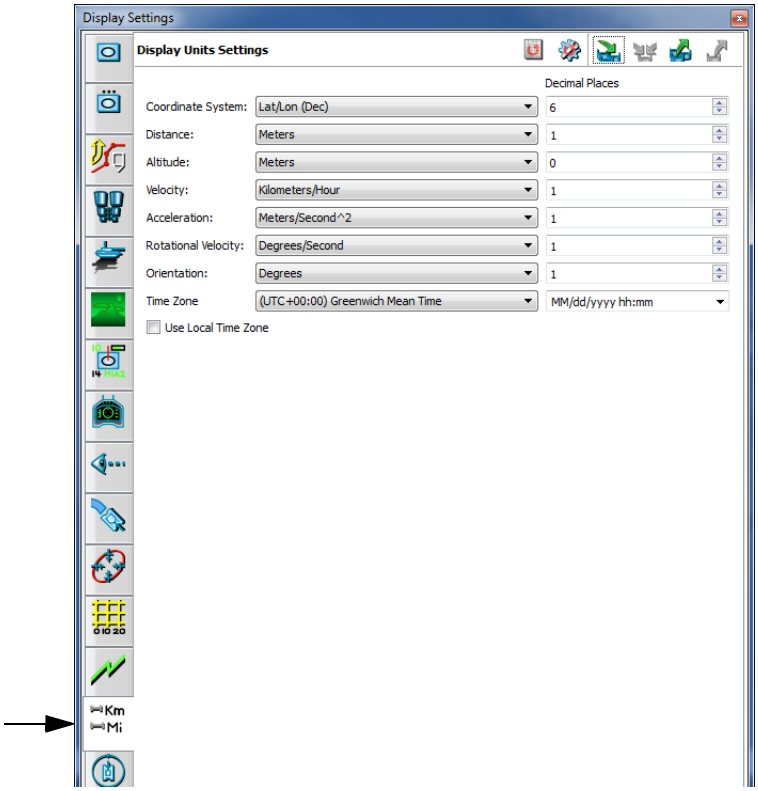


Figure A-8. Display Units Settings page

3. Select the measurement unit you want for each option from its list.
4. Type a precision for each measurement unit in the corresponding box in the Decimal Places column.



B. Command-line Options

VR-Forces has command-line options for vrfGui and vrfSim.

Command-Line Options for vrfGui B-3
Command-line Options for vrfSim B-7
Command-line Options for vrfLauncher B-12
Running in Combined Mode without the Launcher Dialog Box B-12
Using the -- Command-line Argument B-13
Protocol-Independent Command-line Options B-13
Specifying the Site ID and Application Number B-13
Specifying the Language to Use in the GUI B-14
Setting the Notification Level B-14
Loading Plug-ins B-15
Command-line Options for HLA Federations B-15
Specifying the Federation Execution B-15
Specifying the FED File B-16
Specifying the RPR FOM Version B-16
Specifying a FOM Mapper B-16
Specifying the RPR FOM Revision B-16
Specifying FOM Mapper Initialization Data B-17
Specifying a FED File That is Appropriate for the FOM Mapper B-17
Enabling Time Management B-17
Command-line Options for DIS Exercises B-18
Specifying the Port Number B-18
Specifying Point-to-Point or Multicast Operation B-18
Specifying the Multicast Time-to-Live B-18
Using Asynchronous I/O B-18

Specifying the Exercise ID B-19
Specifying the DIS Version B-19

B.1. Command-Line Options for vrfGui

Table B-1 lists the command line options for vrfGui.

Table B-1: vrfGui command-line options (Sheet 1 of 5)

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.
--antiAliasing *level*	Set the anti-aliasing level (0, 2, 4, 8, or 16. Default: 4.
--appDataDir *directory*	Specifies the location of application data.
(-c | --showConsole)	Display a console window.
(-C | --autoConnect) {*filename* | *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. Default: 24.
--dis	Start in DIS mode, using DIS command-line arguments to specify connection parameters.
--dispSetting *filename*	Specifies the display configuration to load.
--doNotCheckPluginVersions	Do not check the version in a plug-in to make sure it can load in VR-Forces. A plug-in gets built against a particular version of VR-Forces. Ordinarily, when VR-Forces loads a plug-in, it checks to be sure it was created with that version of VR-Forces. This option prevents that check.
--doNotLoadVrfPlugins	Prevents loading of plug-ins in the *./plugins* directory.
--entDispSetting *filename*	Specifies the entity display setting to load.
--envSetting *filename*	Specifies the environment configuration file to load.
(-E | --entTypeMap) *filename*	Load the specified entity type mapping file. The file extension determines the type of mapping.
--fileTransporterReceivePort *port*	Port for the file transporter. This is the port used to receive order of battle and plan files.
--fowPerspective *force*	Shows the ground truth and spot reports perspective of the specified force, where *force* can be Friendly, Opposing, Neutral, or None. Default: None.
--fullScreen	Start with the window in full screen mode.

Table B-1: vrfGui command-line options (Sheet 2 of 5)

Option	Description
`--frameRate` *rate*	If rate is 0, the frame rate is variable. Otherwise, specifies a fixed frame rate for rendering.
`(-G | --locale)` *language*	Specifies the locale (for localization).
`--gui` *filename*	Specify a GUI configuration file. This file specifies the layout of menus, toolbars, and dialog box pages. A simple filename assumes *./appdata/settings/vrfGui*. A relative filename is from the *./bin* directory. A full path is used as is.
`(-h | --help)`	Display command-line usage information.
`--hla13`	Start in HLA 1.3 mode, using HLA command-line arguments to specify connection parameters.
`--hla1516`	Start in HLA 1516 mode, using HLA command-line arguments to specify connection parameters.
`--hla1516e`	Start in HLA Evolved mode, using HLA command-line arguments to specify connection parameters.
`(-I | --batchedRendering)`	Disables instance rendering. Default: Disabled.
`(-l | --logFileName)` *filename*	Specifies the log file name.
`(-L | --loadObservers` *filename1 filenamen*	Loads the saved observers in the specified files.
`--loadPlugin` *filename*	Specifies a plug-in to load. Accepted multiple times. The filename can be a DLL or SO or an XML file that specifies the plug-ins to load. The order in which you list multiple plug-ins on the command line does not determine the order in which they are loaded. For details, please see "Managing Plug-ins," on page 5-32.
`(-n | --notifyLevel)` *notify_level*	Specifies the notification level for application messages, where *notification_level* is a number from 0 through 4. The least verbose message level is 0, the highest is 4. The default is 2. For details, please see "Setting the Notification Level," on page B-14.
`(-O | --hasNTPSync)`	Enables synchronized ocean between different VR-Vantage applications, such as VR-Vantage Stealth and the VR-Forces front-end. System clocks must be synchronized.
`--plugin` *filename*	Load the specified plug-in. Can be used multiple times.
`--pvd`	Start the VR-Forces front-end with only 2D modes available.

Table B-1: vrfGui command-line options (Sheet 3 of 5)

Option	Description
(-S | --suppressHatInter-sectOnAttach)	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.
--scenarioFile *filename*	Specifies a scenario to load.
--stealth	Start the VR-Forces front-end with only 3D modes available.
--stencilBits *bits*	Sets the number of stencil bits (0 or 8). Default: 0.
terrain_MTF_file	Specifies a terrain (MTF) file to load.
--unplug *filename*	Do not load the specified plug-in. Can be used multiple times.
--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-Forces may not work properly.	
--userDataDir *directory*	Specifies the user data directory.
(-v | --version)	Display version information and exit.
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.
--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.
--ignoreAdvisories	Enables or disables HLA advisory messages. Default: true.

Table B-1: vrfGui command-line options (Sheet 4 of 5)

Option	Description
(-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.
--rprFomRevision	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, or 2.0017)
(-s | --siteId *ID*	Site ID.
(-S | --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.
--useGeographicDdm	Specifies use of geographic DDM if you are using the MÄK RTI. Default: false.
(-x | --execName) *exec_name*	Execution name. Default: VR-Link.
DIS Only	These options are available if you specify --dis.
(-a | --appNumber) *number*	Application number.
(-A | --destAddrString) *address*	Destination address.
--deviceAddress *address*	Specifies the address of the network card to use for UDP traffic.
--disVersion	The DIS version. Default: 5.
(-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.
--mcastTtl	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.
(-s | --siteId *ID*	Site ID.
--sendBufferSize *size*	Send buffer size. Default: -1.
--subnetMask *mask*	Specifies a subnet mask.

Table B-1: vrfGui command-line options (Sheet 5 of 5)

Option	Description
`--useAsyncIO`	Use asynchronous I/O. Default: False.
`--useIpv6`	Use the IPV6 protocol, if available.
`(-x | --exerciseId) ID`	Exercise ID.

B.2. Command-line Options for vrfSim

Table B-2 describes the command-line options for vrfSim.

Table B-2: vrfSim command-line options (Sheet 1 of 5)

Option	Description
`(-a | --appNumber) application_number`	Specifies the application number. For details, please see "Specifying the Site ID and Application Number," on page B-13.
`--additionalSystemScriptsDirectory directory`	Specifies a directory where VR-Forces will look for system scripts when it populates the Scripted Task dialog box script list.
`--appDataDir directory`	Specifies the location of the appData directory.
`--appIdRange lower-upper`	Specifies a range of application IDs that represent objects generated by VR-Forces.
`(-B | --batchScenarioFileName) batch_file`	Runs the specified batch file. (For use with one back-end only.) For more information, please see *Running in Batch Mode from the Command Line*, under Section 1.6.4, "Running VR-Forces in Batch Mode," in *VR-Forces Scenario Management Guide*.
`(-c | --frontEndPID) PID`	The process ID for the front-end when running in combined mode.
`--countryCodesMappingFile file_name`	Specifies an alternative file that maps country codes to country names and DIS enumeration values.
`(-d | --settingsFile) file_name`	Specifies a configuration file to load instead of *vrfSimSettings.xml*.
`--daemon`	Starts that back-end as a daemon process (Linux only). If you start *vrfSim* as a daemon, it does not try to read characters from the console. It forks a background process at startup. It will exit cleanly if it receives a SITGTERM, SIGQUIT, or SIGINT signal.
`--defaultObjectTimeout timeout`	Specifies the timeout interval for objects.
`--diGuyAnimationsFile file_name`	Specifies an additional configuration file for DI-Guy animations.

Table B-2: vrfSim command-line options (Sheet 2 of 5)

Option	Description
`--diGuyCharacterDataFile` *file_name*	Specifies an additional configuration file for DI-Guy character data.
`--doNotLoadVrfPlugins`	Do not load any VR-Forces plug-ins from the *./plugins* directory.
`--dynamicFeaturesEnabled`	Enables the dynamic features system. If enabled, simulation objects are mapped into the terrain feature system and are part of feature query results. Default: disabled. Enabling dynamic features can decrease performance.
`--frameRateStatisticsLog-File` *file*	Specifies the log file for frame rate statistics when they are enabled. Default: *./vrfSimFrameRateStatistics.log*.
`--fullyLoadTerrain`	For paged terrains, load all pages.
`(-h | --help)`	Displays help information in a console window.
`--hostAddressString` *address*	The IP address of the host computer. Called Device Address in the Launcher.
`(-i | --sessionId)` *session_ID*	Specifies a session ID. For details, please see "Specifying a Session ID," on page 5-8.
`(-L | --scenarioFileName)` "*scenario_pathname*"	Loads the specified scenario. The pathname can be absolute or relative from the *./bin* directory. Mutually exclusive with the `-T` option. For details, please see `Section 1.2.2, "Loading a Scenario from the Command Line,"` in *VR-Forces Scenario Management Guide*.
`--loadPlugin` *plugin_file*	Specifies one or more plug-ins to load. (Accepted multiple times.)
`--logFrameRateStatistics`	Specifies that the back-end should log frame rate statistics.
`--msdlSIDCMappingFile` *file_name*	Specifies an alternative file for mapping SIDC codes to DIS enumerations.
`(-n | --notifyLevel)` *notification_level*	Sets the level for messages written to the console or the VR-Forces log file (*vrf.log*), where *notification_level* is a number from 0 through 4. The least verbose message level is 0, the highest is 4. The default is 2. For details, please see "Setting the Notification Level," on page B-14.
`--outputLogFile` *file_name*	The file to use to write out application output.

Table B-2: vrfSim command-line options (Sheet 3 of 5)

Option	Description
`(-q | --doNotUseConsole)`	Specifies that all *vrfSim* output go to the log file rather than the console (quiet mode). If you are using a high level of notification, sending output to the console can degrade performance. Running in quiet mode prevents this degradation of performance. This command only applies to back-ends running on Windows. To create a log file on Linux, redirect the output of vrfSim, for example: `vrfSim > mylog.txt` For more details about log files, please see "The VR-Forces Log Files," on page 5-39.
`(-r | --startInRunMode)`	Starts the back-end in run mode. If you start the back-end without specifying a scenario (with `-L`) or a terrain database (with `-T`), the back-end starts in pause mode.
`(-s | --siteId)` *site_id*	Specifies the site ID. For details, please see "Specifying the Site ID and Application Number," on page B-13.
`--setMainThreadToHighPriority`	Sets the VR-Forces main thread to high priority. When enabled, other threads may be starved if there are insufficient processors available. If running on hardware with limited processor time, this may decrease overall system performance. Windows only.
`--simulationModelSet` *sms*	Specifies the simulation model set. It must be declared separately for front-end and back-end even in combined mode.
`(-T | --terrainDatabase)` *terrain_database*	Specifies a terrain database to load in the back-end. Mutually exclusive with the `-L` option. For details, please see "Working with Multiple Back-ends," on page 3-8 and "Loading a Terrain Database at Startup," on page 5-21.
`--targetFrameRate` *rate*	Specifies the frame rate above which vrfSim will yield CPU time to other processes.
`--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-Forces may not work properly.	
`--disableParallelPreprocess`	Disables parallel processing for components that implement the preprocess call.
`(-v | --version)` *version*	Displays version information.

Table B-2: vrfSim command-line options (Sheet 4 of 5)

Option	Description
`(-- | --ignore_rest)]`	Ignore all of the command-line options that are listed after this argument. This lets you quickly limit the arguments applied if you want to edit a script or batch file without losing the original command line.
HLA Only	
`(-f | --fomMapperLib)` *libname*	FOM Mapper library name.
`(-F | --fedFileName)` *fedfile_name*	FED file name.
`--fomMapperInitData` *data*	FOM Mapper initialization data.
`--fomModules` *module*	Specifies an HLA Evolved FOM module. Can be used multiple times.
`--ignoreAdvisories`	Enables or disables HLA advisory messages. Default: true.
`--mimModule`	Specifies an HLA Evolved MIM module.
`(-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.
`--rprFomRevision`	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, or 2.0017)
`(-S | --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.
`--timeManagement`	Enables time management for the back-end.
`(-x | --execName)` *exec_name*	Execution name. Default: VR-Link.
DIS Only	
`(-A | --destAddrString)` *address*	Destination address.
`--deviceAddress` *address*	Specifies the address of the network card to use for UDP traffic.
`--disVersion`	The DIS version. Default: 5.

Table B-2: vrfSim command-line options (Sheet 5 of 5)

Option	Description
`--mcastTtl`	Specifies the multicast time-to-live. Default: -1.
`(-S | --multicastAddresses)` *address1 ... addressn*	Specifies one or more multicast addresses.
`(-P | --disPort)` *port*	DIS port.
`--recvBufferSize` *size*	Receive buffer size.
`--sendBufferSize` *size*	Send buffer size. Default: -1.
`--subnetMask` *mask*	Specifies the subnet mask. VR-Forces can usually figure out the subnet mask from the IP address. However, if it does not do this correctly, you can explicitly specify the subnet mask.
`--suppressSelfReflect`	Represses self reflection of the DIS exercise connection.
`--useAsyncIO`	Use asynchronous I/O. Default: False.
`--useIpv6`	Specifies that the DIS exercise connection use IPV6.
`(-x | --exerciseId)` *ID*	Exercise ID.

B.3. Command-line Options for vrfLauncher

Table B-3 describes the command-line options for vrfLauncher.

Table B-3: vrfLauncher command-line options

Option	Description
`(-B | --backend)`	Start vrfLauncher for the back-end only.
`(-C | --config)`	Open the Simulation Connections Configuration tool.
`(-F | --frontend)`	Start vrfLauncher for the front-end only.
`(-h | --help)`	Displays help information in a console window.
`--usePredefinedConnection` *connection*	Specify an already defined connection name, such as DIS localhost, so that the vrfLauncher can launch without needing to display the Simulation Connections dialog box.
`(-v | --version)`	Displays version information.
`--`	Pass all arguments after this to vrfGui and vrfSim, except as modified by `F--` or `B--`.
Special Commands for Arguments that Come After `--`	
`F--`	Arguments after `F--` are passed only to the front-end.
`B--`	Arguments after `B--` are passed only to the back-end.
`-debug`	Appends a "d" to vrfGui or vrfSim to run the debug version. Can only be used following `B--` or `F--`. (Windows only)

The use of the `--` command line option is the opposite of how it is used for other MÄK applications.

B.3.1. Running in Combined Mode without the Launcher Dialog Box

If you want to run VR-Forces in combined mode from the command line, there are no command-line arguments for vrfGui or vrfSim to support this. However, you can run vrfLauncher with the `--usePredefinedConnection` argument and VR-Forces will load the specified connection without displaying the Simulation Connections Configuration dialog box. This has the effect of running directly in combined mode from the command line. For example, suppose you want to start in combined mode using the DIS connection. The command would be:

```
vrfLauncher --usePredefinedConnection DIS
```

B.3.2. Using the -- Command-line Argument

The -- command-line argument is used to send additional command-line arguments to the front-end and back-end. This is the opposite of how this argument is used in other MÄK applications.

It has some special sub-arguments that allow you to direct some arguments only to the front-end or back-end. It also has an argument that lets you run vrfGui or vrfSim in debug mode. Figure B-1 illustrates how these arguments are used.

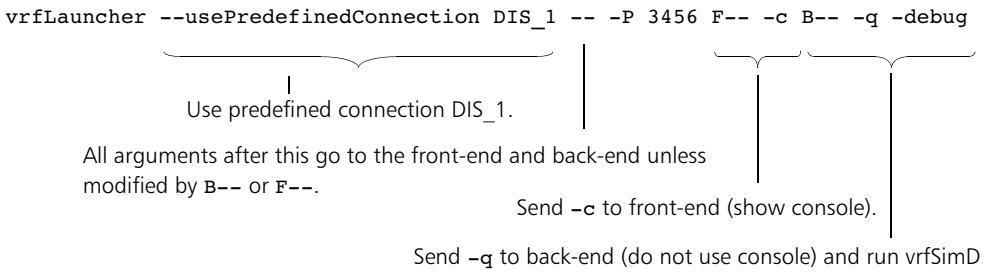


Figure B-1. Using the -- command-line argument

B.4. Protocol-Independent Command-line Options

This section describes in greater detail some of most commonly used command-line options that you can use with both DIS and HLA.

B.4.1. Specifying the Site ID and Application Number

Each front-end and back-end must have a unique site ID:application number pair. To specify the site ID, use the `-s` option. To specify the application number, use the `-a` option, for example:

```
vrfSimDIS -s 2 -a 3005
```

The default application number for vrfGui is 3000; for vrfSim it is 3001.

! If you record a VR-Forces simulation with the MÄK Logger and you want to view the Logger file in VR-Forces, when you view the Logger file, you must run VR-Forces with an application number that is different from the one you used when you recorded the simulation.

B.4.2. Specifying the Language to Use in the GUI

VR-Forces decides which translation files to use based on the language setting for your computer. However, you can override the language setting with the `-G` command-line option, for example, the following command sets the language to German:

```
vrfGui -s 1 -a 3000 -G de --dis
```

If you use this command-line option, you must have a file named *vrf_language_code* (for example, *vrf_de*) in the *./translations* directory. For more information, please see "Localizing the Graphical User Interface," on page 2-11.

B.4.3. Setting the Notification Level

You can control the types of messages sent to the console (and log file on a PC) by specifying a notification level.

➤ To specify a notification level, use the `-n` option, for example:

```
vrfSimDIS -s 1 -a 3001 -n 1
```

The notification levels you can specify are:

- 0 – Only fatal messages are printed.
- 1 – Warning messages and fatal messages are printed.
- 2 – Some diagnostic information is printed with warnings and fatal messages.
- 3 – Verbose information is printed.
- 4 – Debug information is printed.

The default notification level is 2.



The Linux version of VR-Forces does not have a log file. Informational messages are sent to the window from which you start VR-Forces. To capture informational messages, redirect output to a file.

B.4.4. Loading Plug-ins

You can load plug-ins from the command line or by specifying them in a configuration file. (For details about creating a configuration file, please see "Managing Plug-ins," on page 5-32.)

➤ To load a plug-in from the command line, use the `--loadPlugin` command-line option, for example:

```
vrfGui -s 1 -a 3000 --loadPlugin pluginA
  --loadPlugin pluginB --dis
vrfSimDIS -s 1 -a 3001 --loadPlugin pluginA
  --loadPlugin pluginB
```

! Do not specify the extension of the plug-in (for example, *.dll*). If the extension is included, the plug-in will not load.

VR-Forces can display a list of the plug-ins that are loaded. For details, please see "Viewing a List of Loaded Plug-ins," on page 5-38.

B.5. Command-line Options for HLA Federations

VR-Forces listens to whatever port is specified in the RID file. For information about the RID file, consult your RTI documentation.

B.5.1. Specifying the Federation Execution

By default, VR-Forces uses the *VR-Link.fed* file and the VR-Link federation execution.

➤ To specify the HLA federation execution name, use the `-x` option, for example:

```
vrfSimHLA13 -s 1 -a 3001 -x VR-Link20017-1
  --rprFomVersion 2.0017 -F VR-Link20017-1.fed
```

! If you are using the MÄK RTI and are running multiple, concurrent federation executions, you must run the rtiexec. In other words, you cannot use the MÄK RTI in lightweight mode if you are running multiple federations. Please see the MÄK RTI documentation for details about how to configure it to use the rtiexec.

B.5.2. Specifying the FED File

In combined mode, specifying `-F` in the command line also applies to the back-end. If you are running a back-end independently and want to specify a FED file name, you must use `-F`.

➤ To specify a FED file at startup, use the `-F` option.

B.5.3. Specifying the RPR FOM Version

VR-Forces has built-in support for the RPR FOM. The default version is 1.0.

You can specify a different RPR FOM version (0.5, 0.7, 0.8, 2.0006, 2.0014, 2.0017, or 2.0020). If you specify a different FOM version, you must also specify the correct FED file (with `-F`).

```
vrfSimHLA13 -s 1 -a 3001 --rprFomVersion 2.0017
  -F VR-Link20017-1.fed
```



Please see *VR-Forces Release Notes* for a list of the RPR FOM versions supported by your version of VR-Forces.

B.5.4. Specifying a FOM Mapper

VR-Forces supports alternative FOMs through the VR-Link FOM Mapper. You can specify which FOM Mapper to use with the `-f` command-line option:

```
vrfSimHLA13 -s 1 -a 3001 -x VR-Link20017-1
  --rprFomVersion 2.0017 -F VR-Link20017-1.fed
  -f myFomMapper
```

B.5.5. Specifying the RPR FOM Revision

RPR FOM revision 2 for version 2.0017 fixes an encoding and decoding size issue with object ID lists to match RPR FOM 2 draft 17.

➤ To specify the RPR FOM revision, use the `--rprFomRevision` *`revision`* command-line option, for example:

```
vrfSimHLA13 -s 1 -a 3001 --rprFomVersion 2.0017
  -F VR-Link20017-1.fed --rprFomRevision 2
```

B.5.6. Specifying FOM Mapper Initialization Data

Some FOM Mappers require initialization data. If you specify a FOM Mapper that requires such data, specify the data with the `--fomMapperInitData` option:

```
vrfSimHLA13 -s 1 -a 3001 -x VR-Link20017-1
   --rprFomVersion 2.0017 -F VR-Link20017-1.fed
   -f myFomMapper --fomMapperInitData ABC
```

The string "ABC" is passed to the FOM Mapper creation function in the shared library as an initialization parameter.

B.5.7. Specifying a FED File That is Appropriate for the FOM Mapper

The `-f` option only tells the application which FOM Mapper to use. It does not ensure that you are using the right FED file for the FOM Mapper. Make sure that you are using a FED file that is consistent with the FOM Mapper you have chosen.

By default VR-Forces uses a FED file called *VR-Link.fed*, but you can specify an alternate FED file using the `-F` option.

The following example instructs the application to use the RPR FOM 2.0, version 17 FOM mapping code, in the federation execution VR-Link20017, with the *VR-Link20017-1.fed* file:

```
vrfSimHLA13 -s 1 -a 3001 --rprFomVersion 2.0017
   -x VR-Link20017 -F VR-Link20017-1.fed
```

B.5.8. Enabling Time Management

You can enable time management for the back-end from the command line, as follows:

```
vrfSimHLA13 -s 1 -a 3001 --timeManagement -F VR-LinkTM.fed
   -x VR-LinkTM
```

B.6. Command-line Options for DIS Exercises

This section describes command-line options that apply only to using VR-Forces with DIS exercises.

B.6.1. Specifying the Port Number

By default, VR-Forces listens to port 3000.

➤ To change the port that VR-Forces listens to, use the `-P` option, for example:

```
vrfSimDIS -s 1 -a 3001 -P 3500
vrfGui -s 1 -a 3000 --dis -P 3500
```

B.6.2. Specifying Point-to-Point or Multicast Operation

You can specify that VR-Forces run in point-to-point mode or multicast.

➤ To specify point-to-point, use the `-A` option, for example:

```
vrfSimDIS -s 1 -a 3001 -A 123.123.12.1
```

➤ To specify multicast, use the `-S` option, for example:

```
vrfSimDIS -s 1 -a 3001 -S 233.233.22.3
```

You can specify more than one multicast address, as follows:

```
vrfSimDIS -s 1 -a 3001 -S address -S address ...
```


The default is broadcast mode.

B.6.3. Specifying the Multicast Time-to-Live

The multicast time-to-live specifies the number of routers that a message can pass through when PDUs are sent to multicast addresses.

➤ To specify the multicast time-to-live value, use the `--mcastTtl` option, for example:

```
vrfSimDIS -s 1 -a 3001 -S address -S address --mcastTtl 3
```

B.6.4. Using Asynchronous I/O

If you want to use asynchronous I/O for DIS exercises, start VR-Forces with the `--useAsyncIO` option. Using asynchronous I/O can help prevent dropped packets and the resulting loss of information. There is no comparable command-line option for HLA. To use asynchronous I/O in HLA, you set a RID file parameter.

B.6.5. Specifying the Exercise ID

➤ To set the VR-Forces exercise ID, use the **-x** option, for example:

```
vrfSimDIS -s 1 -a 3001 -x 6
```

B.6.6. Specifying the DIS Version

➤ To specify the DIS version, use the **--disVersion** command-line options, for example:

```
vrfSimDIS -s 1 -a 3001 --disVersion 6
```

Please see *VR-Forces Release Notes* for a list of supported versions.



MÄK Products Glossary

This glossary defines terms used in MÄK product documentation.

absolute dead reckoning

A method of dead-reckoning in which an application uses the time stamps contained within state updates if the sender is using absolute time stamping. Contrast with relative dead reckoning.

aggregate

The combination of individual entities, aggregates, or both into a single object. For example, an organizational group such as a platoon.

API

Application Programming Interface.

articulated part

A part of an entity that is capable of movement relative to the entity, such as a turret or gun.

attached part

An object that is attached to another object, such as a rocket attached to a jet.

body coordinate frame

A coordinate framework centered on an entity. To represent the orientation of an entity, an application uses Euler angles or a rotation matrix that rotates from a reference frame to the body coordinate frame.

channel

A channel renders the view of an observer in a 3D visualization application.

clipping

A feature that prevents the display of terrain and objects or parts of objects, that are closer than or farther than specific distances from the observer.

clipping planes

The range of distances from the observer (near clipping and far clipping) in which an application clips objects.

culling

The process of discarding database objects which are not within view, and therefore need not be rendered and displayed.

Cartesian coordinates

A system for indicating location by means of three planes intersecting at right angles to one another at the origin.

Data Distribution Management

The set of HLA services used to specify the routing of data between publishers and subscribers. DDM adds greater control over the Declaration Management (DM) services that only specify filtering based on the class of the data (e.g., ground vehicle versus aircraft). DDM uses the data content to express regions of interest and regions of affect whose overlap establish a communication channel. An example is geographic filtering where a publisher indicates a region around the entity or effect and a subscriber indicates a region expressing its sensor range.

dead-reckoning

A process by which an application calculates the expected location of an entity during periods between state updates, based on velocity, acceleration, and rotational velocity.

DDM

Data Distribution Management.

Defense Modeling and Simulation Office

The former name of the executive secretariat for the Executive Council on Modeling and Simulation (EXCIMS), which provided a full-time focal point for information concerning Department of Defense modeling and simulation (M&S) activities. Renamed Modeling and Simulation Coordination Office. Currently the MSCO promulgates M&S policy, initiatives, and guidance to promote cooperation among DoD components to maximize efficiency and effectiveness.

DIS

Distributed Interactive Simulation.

DIS data packets

See Protocol Data Unit.

display engine

An object in an application that can render 3D data.

Distributed Interactive Simulation

The DIS protocol is a set of standards that govern how participating applications share information about the virtual world. The protocol specifies a set of packets, called protocol data units (PDUs), that communicate this information. Each PDU identifies the sender and contains other information depending on the PDU type. The DIS protocol also specifies when and how frequently PDUs are sent.

The DIS protocol has been superseded by the High-Level Architecture for use by the Department of Defense.

DMSO

See Defense Modeling and Simulation Office.

element definition

In VR-Forces and VR-Vantage, an element definition specifies all of the visualizers that control the various aspects of visualizing a scene element. For example, an element definition includes the main model or military symbology, trailing effects, cockpit displays, and so on.

emitter

A simulated device on an entity that emits electromagnetic radiation, for example, radar.

entity

An element in a simulation, such as a vehicle or a person, that is represented in the simulation through the issuance of state messages.

entity maintenance

A process by which the MÄK Data Logger compensates for discontinuities following a time jump. The Logger sends out interim state messages for entities that were present before the time jump so that they do not time out before the next update message arrives.

entity-type

A list of seven components as specified by the DIS protocol and the HLA RPR FOM. If none of the seven components contains a wildcard (-1) value, then the entity type refers to a specific, narrowly-defined entity. If some components contain a wildcard, then the entity type refers to a class of entities. A larger number of wildcards indicates a broader, more general class.

The components of an entity-type are:

- Entity kind
- Domain
- Country
- Category
- Subcategory
- Specific
- Extra.

environmental process

According to the IEEE 1278.1a specification (DIS), environmental process PDUs communicate simple environment variables, small scale environmental updates, and embedded processes.

Euler angles

A set of three angles used to describe the orientation of an entity as a set of three successive rotations about three different orthogonal axes (x, y, and z). These angles specify successive rotations needed to transform from a reference coordinate system to the entity's body coordinate system.

event

An interaction between objects or between an object and the terrain, such as firing of a munition, or a collision of entities.

exercise

The DIS term for a one or more interacting simulation applications. Compare to federation execution in HLA.

exercise connection

The object (*DtExerciseConn*) through which a VR-Link application connects to the network. State messages are sent through the exercise connection and information about remote entities is received through the exercise connection. (All MÄK products are VR-Link applications.)

exercise ID

A numeric identification for a DIS simulation exercise.

FED file

Federation Execution Data file. The Federation Execution Data is the subset of FOM data needed and read by the RTI. VR-Link applications also read the FED file.

federate

A connection to the RTI. Typically a single simulation application can be thought of as a federate.

federation

A group of HLA federates capable of playing in the same federation execution.

Federation Object Model

Defines the data content of a federation execution.

federation execution

The federation execution represents the actual operation, over time, of a subset of the federates and the RTI initialization data taken from a particular federation. A federation execution is the logical equivalent of a DIS simulation exercise.

field of view

Controls the perspective of the observer.

A wide field of view creates an effect like that of a wide-angle camera lens. Objects appear smaller and farther away from the observer, since the observer coverage spans a wider area. Depth become exaggerated.

A narrow field of view creates an effect like that of a telephoto lens. Objects appear larger and closer to the observer, and the overall scene depth appears flattened. The distances between objects appears compressed.

filter range

A setting that prevents distant entities from being processed while allowing distant terrain to appear normally.

FOM

See Federation Object Model.

FOM Module

In HLA Evolved, defines additional modular data content for a federation execution, usually extensions to an existing FOM. (Also supported by the HLA 1.3 and HLA 1516 versions of the MÄK RTI. Per the HLA Evolved interface specification:

"A partial FOM (containing some or all OMT tables) that specifies a modular component of a FOM. A FOM module may contain classes not inherent to it but upon which the FOM module depends, i.e., superclasses to the modular components. These superclasses will be included in the FOM module either as complete or scaffolding definitions."

frame rate

The rate at which the application displays updated images.

ground clamping

A process by which the a 3D visualization application keeps an entity anchored to the surface of its terrain database, regardless of the altitude data contained in its entity state message.

geocentric coordinates

A coordinate system calculated with respect to the earth's center. The origin of the geocentric coordinate system is the center of the earth. The positive X-axis passes through the prime meridian at the equator; the positive Y-axis passes through 90 degrees east longitude at the equator; and the positive Z-axis passes though the north pole.

geodetic coordinates

A coordinate system in which position is determined relative to a reference ellipsoid, such as the surface of the earth at sea level. In MÄK applications, geodetic coordinates consist of latitude and longitude in radians, and altitude in meters above the reference ellipsoid.

gridded data

Data that has been processed in a rectangular array of points, in X, Y or latitude/longitude, at which single data values define a two dimensional function. According to the IEEE 1278.1a specification, gridded data transmits information about large-scale or high-fidelity spatially and temporally varying ambient fields and about environmental processes and features.

guise

An alternative entity type used to display an object depending on the force ID. For example, a tank could look like an M1A1 to friendly forces and a T72 to hostile forces.

head-up display

A set of indicators and readouts superimposed onto a graphics display. Also called an overlay.

heartbeat

In DIS, the frequency with which current PDUs are sent to the network regardless of whether or not the entity's state has changed.

High-Level Architecture

The High Level Architecture (HLA) for simulations is a U. S. Department of Defense (DOD)-wide initiative to provide an architecture to support interoperability and reuse of simulations. The HLA supersedes DIS for the DOD.

HLA

See High-Level Architecture.

interaction

A message describing a simulation event. An interaction describes an event, it does not update an object's state.

Logger control PDUs

A set of DIS PDUs or HLA interactions that let you control the MÄK Logger remotely.

logical range

A suite of TENA Resources, sharing a common object model, that work together for a given range event. Similar in concept to the HLA Federation.

Local RTI Component

The libraries on a computer that implement an RTI. A federate communicates with the HLA network through the LRC.

LRC

See Local RTI Component.

MÄK Technologies Lisp

An adaptation of the Lisp language used in configuration files for MÄK products.

message

A general term used to refer to DIS PDUs and HLA interactions and state updates. In TENA, a message is similar to an HLA interaction.

MIM

The MOM & Initialization Module (MIM) allows for extensions to be made to the HLA standard MOM. It is a subset of the FOM that contains OMT tables that describe the HLA MOM. The MIM also contains additional predefined HLA constructs such as object and interaction roots, data types, transportation types and dimensions. HLA specifies a standard MIM that is incorporated into all FDDs automatically by the RTI. The standard MIM can be replaced with a user-supplied MIM containing the standard MIM plus extensions.

model definition

In VR-Forces and VR-Vantage, a model definition specifies the 3D model and other basic attributes of a model. It is referenced by an element definition.

Modeling and Simulation Coordination Office

The executive secretariat for the Executive Council on Modeling and Simulation (EXCIMS), which provided a full-time focal point for information concerning Department of Defense modeling and simulation (M&S) activities. The MSCO promulgates M&S policy, initiatives, and guidance to promote cooperation among DoD components to maximize efficiency and effectiveness. (Formerly DMSO)

MSCO

Modeling and Simulation Coordination Office. (Formerly DMSO.)

MTL

See MÄK Technologies Lisp.

orientation clamping

Adjusts the pitch and roll of an entity so that it appears properly seated when the terrain is inclined. For example, if a tank is moving horizontally across the face of a hill, orientation clamping prevents the tank from appearing level, and therefore, partially embedded into the hillside. Used with ground clamping.

object

An element in a simulation that has persistence, as opposed to an interaction, which is a transient element.

object handle

An integer that an application uses to identify a particular object in RTI service calls. An object handle is meaningful only to a particular federate. The same object can be known to different federates by different object handles.

object name

A character string that can be used to identify an object. In HLA, the object name is known to the RTI, and the RTI provides functions to find out an object's name, given its handle, and vice versa. Object names can be chosen by applications that register the objects with the RTI, however if you do not want to choose names for objects, the RTI will assign names for you.

observer

In MÄK 3D applications, the point of view into the scene. (Called the eyepoint in MÄK Stealth 6.x and earlier.)

PDU

See Protocol Data Unit.

Protocol Data Unit

A unit of data message (packet) that is passed on a network between DIS simulation applications.

radio transmitter

A simulated device on an entity, capable of transmitting radio communications.

radio receiver

A simulated device on an entity, capable of receiving radio communications.

Realtime Platform Reference FOM

An HLA reference FOM based on the DIS protocol.

recording

A Data Logger file that stores a history of the interactions in a simulation for playback.

reflected entity list

A list of entities simulated by remote applications (federates in HLA) and about which the local simulation has received information over the network.

reference FOM

A FOM designed to be used as a whole, or with modification, by a wide variety of similar federation executions.

relative dead reckoning

A method of dead reckoning in which an application approximates the location of an object based on the local time of receipt of the last state update message.

remote display engine

A display engine running in an application that is separate from the master application, and which is controlled by the master application.

RID file

See RTI Initialization Data.

RTI Initialization Data

The initialization data required by the RTI for operation.

Data required by an RTI during initialization, independent of the FOM being used. RID data is usually dependent on a specific implementation of the RTI.

RPR FOM

See Realtime Platform Reference FOM.

RTI

See Run-Time Infrastructure.

Run-Time Infrastructure

A library and other supporting software that implements the HLA interface specification. All federates communicate with one another in an HLA environment through RTI functions.

scene element

In VR-Vantage and VR-Forces, any object that is rendered in the scene, such as an entity model, interaction, or tactical graphic.

Simulation Interoperability Standards Organization

A group that seeks to promote modeling and simulation interoperability and reuse for the benefit of diverse M&S communities, including developers, procurers, and users, world-wide.

simulation time

In VR-Forces, simulation time is used in dead-reckoning of remote entities and thresholding of local entities. Typically, simulation time is set once during each iteration of the application's main simulation loop so that all entities are dead-reckoned based on the same value of current time.

SISO

See Simulation Interoperability Standards Organization.

smooth period

The period of time over which trajectory smoothing takes place.

smoothing

A method of ensuring that transitions from an entity's dead-reckoned position to its actual position are not so abrupt as to be visually disconcerting.

state

The current status of an object, including location, direction of movement, extent of damage, and so on.

tape

An alternative term for a Logger recording. Not a physical magnetic tape.

terrain following

In a 3D visualization application, causes the observer (eyepoint) to maintain a constant distance above the terrain surface. The observer's height changes in tandem with the peaks and valleys as it passes over the geography.

timeout

The period of time in which an application continues to display an entity after that entity's update messages have stopped appearing on the network. Time-outs are usually not used in HLA because there is no set frequency (no heartbeat) for transmitting messages.

timescale

A factor by which time is accelerated or slowed during playback of a Data Logger recording.

topographic coordinates

A right-handed Cartesian coordinate system whose X-Y plane is tangent to the earth's surface at the origin, with the positive X axis pointing north, the positive Y axis pointing east, and the positive Z axis pointing down. There are an infinite number of topographic coordinate systems – one for each point on the earth's surface.

topographic coordinate frame

A coordinate frame in the context of the terrain.

trajectory smoothing

A method used by to smooth positional discontinuities that could occur when new state updates arrive.

UDP port

A network channel through which an application sends and receives data for DIS exercises.

Universal Transverse Mercator

In general, a non-cartesian coordinate system in which the X, Y, and Z correspond to easting (nearly east), northing (nearly north), and height above an ellipsoid that approximates the surface of the earth.

UTM

See Universal Transverse Mercator.

view control messages

A set of programmatic messages that let you control the view in VR-Vantage applications remotely.



Index

-- command-line option B-3, B-10, B-12
--additionalSystemScriptsDirectory command-line option B-7
--antiAliasing command-line option B-3
--appDataDir command-line option B-3, B-7
--appIdRange command-line option B-7
--appNumber command-line option B-5, B-6, B-7
--autoConnect command-line option B-3
--backend command-line option B-12
--batchedRendering command-line option B-4
--config command-line option B-12
--countryCodesMappingFile command-line option B-7
--daemon command-line option B-7
--dataDir command-line option B-3
--defaultObjectTimeout command-line option B-7
--depthBuffer command-line option B-3
--destAddrString command-line option B-6, B-10
--deviceAddress command-line option B-6, B-10
--diGuyAnimationsFile command-line option B-7
--diGuyCharacerDataFile command-line option B-8
--dis command-line option B-3
--disableParallelPreprocess command-line option B-9
--disPort command-line option B-6, B-11
--dispSetting command-line option B-3
--disVersion command-line option B-6, B-10, B-19
--doNotCheckPluginVersions command-line option B-3
--doNotLoadVrfPlugins command-line option B-3, B-8
--dynamicFeaturesEnabled command-line option B-8
--entDispSetting command-line option B-3
--entTypeMap command-line option B-3
--envSetting command-line option B-3
--execname command-line option B-6, B-10
--exerciseId command-line option B-7, B-11
--federateName command-line option B-6, B-10
--federateType command-line option B-6, B-10
--fedFileName command-line option B-5, B-10
--fileTransporterReceivePort command-line option B-3
--fomMapperInitData command-line option B-5, B-10
--fomMapperLib command-line option B-5, B-10
--fomModules command-line option B-5, B-10
--fowPerspective command-line option B-3
--frameRate command-line option B-4
--frameRateStatisticsLogFile command-line option B-8
--frontend command-line option B-12
--frontEndPID command-line option B-7
--fullScreen command-line option A-6, B-3
--fullyLoadTerrain command-line option B-8
--gui command-line option B-4
--help command-line option B-4
--hla13 command-line option B-4
--hla1516 command-line option B-4
--hla1516e command-line option B-4
--hostAddressString command-line option B-5, B-6, B-8
--ignore_rest command-line option B-3, B-10
--loadObservers command-line option B-4
--loadPlugin command-line option B-4, B-8, B-15
--locale command-line option B-4

--localSettingsDesignator command-line option B-6, B-10
--logFileName command-line option B-4
--mcastTtl command-line option B-6, B-11
--mimModule command-line option B-10
--msdlSIDCMappingFile command-line option B-8
--multicastAddresses command-line option B-6, B-11
--notifyLevel command-line option B-4, B-8
--outputLogFile command-line option B-8
--plugin command-line option B-4
--pvd command-line option B-4
--recvBufferSize command-line option B-6, B-11
--rprFomRevision command-line option B-6, B-10
--rprFomVersion command-line option B-6, B-10
--scenarioFile command-line option B-5
--sendBufferSize command-line option B-6, B-11
--sessionId command-line option B-5, B-6
--setMainThreadToHighPriority command-line option B-9
--settingsFile command-line option, vrfSim B-7
--showConsole command-line option B-3
--simulationModelSet command-line option B-9
--siteId command-line option B-6
--stealth command-line option B-5
--stencilBits command-line option B-5
--subnetMask command-line option B-6, B-11
--suppressHatIntersectOnAttach command-line option 12-5, B-5
--suppressSelfReflect command-line option B-11
--synchronizeOcean command-line option B-4
--targetFrameRate command-line option B-9
--timeManagement command-line option 5-18, B-10, B-17
--unplug command-line option B-5
--useAbsoluteTimeStamps command-line option B-5, B-9
--useAdvisories command-line option B-5, B-10
--useAsyncIO command-line option B-7, B-11, B-18
--useGeographicDdm command-line option B-6
--useIpv6 command-line option B-7, B-11
--userDataDir command-line option B-5
--version command-line option B-5, B-9, B-12
-debug command-line option B-12, B-8
B-- command-line option B-12, B-8
F-- command-line option B-12, B-8

Numerics

2D
  icon
    changing size 7-14
    label 7-4
    scaling 7-14
  label, shadow text 7-9
  observer, attachment behavior 12-14
  observer coordinate system 11-7
  projection, switching to 3D 4-3
  window 5-13
2D Icon model set 7-15
3D
  model, licensing 1-22
  observer coordinate systems 11-6
  projection, switching to 2D 4-3
  window 5-13
3D Colorized Models model set 7-15
  color, changing 7-19
3D Models model set 7-15
3DS 4-8
  file 3-32

A

-A command-line option B-6, B-10, B-18
-a command-line option B-5, B-6, B-7, B-13
abandoning, plan 3-28
Absolute mode 12-6
acceleration
  displaying 7-4
  specifying measurement unit A-9
actuator 3-17, 3-18
adding
  extended label 7-11
  feature layer 3-33
  observer 10-5, 10-6
  plug-in 5-35
  text, to title bar A-8
adjusting, sound volume 15-3
advanced lighting and shading 17-4
aggregate
  aggregate-level 3-24
  bounding volume, color 9-11
  closing formation 3-23
  collapsing 9-3
  collapsing to root 9-3
  count of 6-23
  disaggregated and aggregated 3-20

aggregate (continued)
displaying by level 9-4
displaying ghosted entities 9-6
echelon ID 3-22
entity-level 3-20
expanded and collapsed 3-21
expanding 9-3
expanding all 9-3
expanding and collapsing 9-2
formation 3-23
information 6-15, 6-20
movement 3-23
organization concepts 3-22
reorganizing 3-22, 3-23
sector 7-21
selecting 9-2
task behavior 3-24
Aggregate Display Settings page 8-15, 8-18, 9-7, 9-10, 9-11
Aggregate Information dialog box 6-15
aggregated aggregate 3-20
aggregate-level modeling 3-11, 3-20
AggregateLevel.sms 3-11
agility, terrain 1-12
altitude 7-3
entity, displaying 7-20
intervisibility line 14-10
specifying measurement unit A-9
switching from 2D to 3D 4-3
vertex, displaying 6-14
altitude point
changing color 18-6
deleting 18-5
inserting 18-5
Always Join Session with Open Database 5-17
Always Join Session with Session Database 5-17
ambient
color 17-10
value, material 17-10
ambient occlusion map 1-15
ammunition, status 6-16
animation
attack 8-15
DI-Guy B-7
observer view transition 11-25
anti-aliasing B-3
appData directory B-7
appearance
bit 17-3
DI-Guy B-7
appending, saved views to view list 11-26
application
data, specifying location of B-3
ID 3-5
number B-5, B-6, B-7
default B-13
specifying at startup B-13
Application Settings dialog box
GUI Layout page A-4
Mouse Mappings page 11-13
Performance Options page 8-6
Session Options page 5-7, 11-19
architecture, component 3-17
area 3-25, 3-27
disaggregation 3-25
feature, list of 18-14
terrain page-in 3-25
articulated part, attaching to 12-4
Ask to Join Session 5-17
assertOnBlockingTerrainCalls parameter 3-37
assigning
echelon ID 3-16
session ID 5-8
asynchronous I/O B-18
attach 12-6
observer, articulated part 12-4
attach mode 1-14, 11-3, 11-7, 12-6
2D behavior 12-14
Absolute 12-6
Follow 12-7
Mimic 12-8
Mimic Track 12-9
movement, attached context 11-7
prop, affect of 12-5
Space Follow 12-10
Tether 12-11
Track 12-13
attached, context 11-7
attaching observer
to entity or prop 12-2
to object, with keyboard 12-3
attachment
list, filtering 12-4
order 12-3
preservation of view changes 12-3
Attachment Panel 12-2
filtering attachment list 12-4
attack
animation 8-15
indicator 8-15

audio 15-2
Audio Settings dialog box 15-2, 15-3, 15-5, 15-6, 15-7, 15-8
Audio Settings page 15-2, 15-3
Automatically Join Session 5-16
Automatically Open Session Database without Joining Session 5-16

B

-B command-line option B-7, B-12
back-end 3-3
  dropping from exercise 3-8
  enabling time management 5-18, B-17
  feature representation 3-34
  late joining 3-8
  missing 3-9
  multiple 3-3, 3-8
  remapping 3-9
  sending messages 3-4
  starting 5-3
    as daemon B-7
  synchronizing 3-41
background
  2D label 7-9
  intervisiblity information 14-10
batch file, command-line option B-7
bearing, intervisibility line 14-10
behavior, entity 3-19
berkeley, database 18-17
B-HAVE Module 1-17
bit, depth B-3
bitmap 3-33
  location of 2-4
blur 16-2, 16-4
BMP 3-33
body coordinates 11-3
border, icon 7-13
bounding box, selecting objects 6-5
bounding volume
  aggregate, color 9-11
  entities 7-27, 7-28
Brains for Human Activity in Virtual Environments 1-17
brightness 16-4
broadcast, stealth 10-9
broadcasting 10-8
buffer, size B-6, B-11
bump map 1-15
buoyancy
  enabling or disabling 8-36
  LOD distance 8-36
button, toolbars 5-11

C

-C command-line option B-3, B-12
-c command-line option B-7
caching, terrain data 1-19
call, preprocess B-9
camera
  effects 16-2
  moving 11-12
  ways to move 11-11
category, echelon 3-15
changing
  color
    altitude point 18-6
    force 7-19
  extended label index 7-12
  gain 11-23
  icon size 7-14
  intervisibility object 14-7
  model set 7-15
  observer mode 10-2
  unit of measurement A-9
channel
  keyword, compass 11-5
  observer 10-5
  sound mapping 15-8
character, DI-Guy B-7
checking
  version, plug-in B-3
clamping, entities 8-3
clearing
  object console 7-32
  view list 11-25
clip
  far, ocean planar reflection 17-7
clipboard 7-31
clock mode 3-40
closing
  all windows 5-24
  application 5-24
  formation 3-23
  simulation engine 5-24
cloud
  disabling 17-15
  skybox cube map 17-12

cockpit display 8-27
  enabling or disabling 8-25
Cockpit Display Settings page 8-26
Collada 3-32
collapsing
  aggregate 9-2, 9-3
    to root 9-3
color
  2D label background 7-9
  aggregate bounding volume 9-11
  altitude point
    changing 18-6
    deleting 18-5
    inserting 18-5
  ambient 17-10
  compass 11-5
  diffuse 17-10
  entity label 7-3
  force, changing 7-19
  intervisibility, configuring 14-8
  radio communication line 8-23
  specular 17-10
  squawk indicator 8-23
  terrain, configuring 18-5
coloring, terrain 18-4
combat
  aggregate, graphics 8-15
combined mode, vrfLauncher B-12
command line
  loading terrain B-5
  options B-3
command-line option B-1
  -- B-3, B-10, B-12
  -A B-6, B-10, B-18
  -a B-5, B-6, B-7, B-13
  --additionalSystemScriptsDirectory B-7
  --antiAliasing B-3
  --appDataDir B-3, B-7
  --appIdRange B-7
  --appNumber B-5, B-6, B-7
  --autoConnect B-3
  -B B-7, B-12
  B-- B-12, B-8
  --backend B-12
  --batchedRendering B-4
  -C B-3, B-12
  -c B-7
  --config B-12
  --countryCodesMappingFile B-7
  --daemon B-7
command-line option (continued)
  --dataDir B-3
  -debug B-12, B-8
  --defaultObjectTimeout B-7
  --depthBuffer B-3
  --destAddrString B-6, B-10
  --deviceAddress B-6, B-10
  --diGuyAnimationsFile B-7
  --diGuyCharacerDataFile B-8
  --dis B-3
  --disableParallelPreprocess B-9
  --disPort B-6, B-11
  --dispSetting B-3
  --disVersion B-6, B-10, B-19
  --doNotCheckPluginVersions B-3
  --doNotLoadVrfPlugins B-3, B-8
  --dynamicFeaturesEnabled B-8
  -E B-3
  --entDispSetting B-3
  --entTypeMap B-3
  --envSetting B-3
  --execname B-6, B-10
  --exerciseId B-7, B-11
  -F B-4, B-12
  F-- B-12, B-8
  -f B-5, B-10, B-16
  --federateName B-6, B-10
  --federateType B-6, B-10
  --fedFileName B-5, B-10
  --fileTransporterReceivePort B-3
  --fomMapperInitData B-5, B-10
  --fomMapperLib B-5, B-10
  --fomModules B-5, B-10
  --fowPerspective B-3
  --frameRate B-4
  --frameRateStatisticsLogFile B-8
  --frontend B-12
  --frontEndPID B-7
  --fullScreen A-6, B-3
  --fullyLoadTerrain B-8
  -G B-4, B-14
  --gui B-4
  -H B-5, B-6
  -h B-4, B-8, B-12
  --help B-4
  --hla13 B-4
  --hla1516 B-4
  --hla1516e B-4
  --hostAddressString B-5, B-6, B-8
  -I B-4

command-line option (continued)
-i 5-8, B-5, B-6, B-8
--ignore_rest B-3, B-10
-L B-4, B-8
-l B-4
--loadObservers B-4
--loadPlugin B-4, B-8, B-15
--locale B-4
--localSettingsDesignator B-6, B-10
--logFileName B-4
--mcastTtl B-6, B-11
--mimModule B-10
--msdlSIDCMappingFile B-8
--multicastAddresses B-6, B-11
-N B-5, B-10
-n B-4, B-8, B-14
--notifyLevel B-4, B-8
-O B-4
--outputLogFile B-8
-P B-6, B-11, B-18
-p B-6, B-10
--plugin B-4
--pvd B-4
-q B-9
-r B-9
--recvBufferSize B-6, B-11
--rprFomRevision B-6, B-10
--rprFomVersion B-6, B-10
-S 12-5, B-5, B-6, B-10, B-11, B-18
-s B-9, B-13
--scenarioFile B-5
--sendBufferSize B-6, B-11
--sessionId B-5, B-6
--setMainThreadToHighPriority B-9
--settingsFile B-7
--showConsole B-3
--simulationModelSet B-9
--siteId B-6
--stealth B-5
--stencilBits B-5
--subnetMask B-6, B-11
--suppressHatIntersectOnAttach 12-5, B-5
--suppressSelfReflect B-11
--synchronizeOcean B-4
-T B-9
--targetFrameRate B-9
--timeManagement 5-18, B-10, B-17
--unplug B-5
--useAbsoluteTimeStamps B-5, B-9
--useAdvisories B-5, B-10
command-line option (continued)
--useAsyncIO B-7, B-11, B-18
--useGeographicDdm B-6
--useIpv6 B-7, B-11
--userDataDir B-5
-v B-5, B-9, B-12
--version B-5, B-9, B-12
-x B-6, B-7, B-10, B-11, B-15
DIS B-19
communication
component 3-17
line 8-20
line-of-sight 8-22
communication model
custom 3-16
entity 3-16
external 3-16
compass
color 11-5
configuring 11-5
location 11-5
model definition 11-5
observer 11-4
component
actuator 3-18
architecture 3-17
communication among 3-17
controller 3-17, 3-18
sensor 3-17
composing, terrain 3-31
concurrent task 3-28
configuration
entity display B-3
environment B-3
file, RTI 2-10
plug-in, deleting 5-37
configuring
compass 11-5
contour lines 18-10
depth of field 8-34
display B-3
GDB soil type 18-2
grid lines 18-8
ground clamping 8-4
instrument panel 8-28
intervisibility objects 14-8
ocean planar reflection 17-7
orientation clamping 8-4
props 3-33
range rings 7-24

configuring (continued)
sensor effects 16-4
session 5-16
shadows 8-31, 17-15
system for MÄK RTI 2-10
terrain color settings 18-5
terrain profile 13-3
time management 5-17
track history 8-12
connecting, to network automatically B-3
connection, information 5-31
Connection Information dialog box 5-31
console
messages, sent to B-14
object 7-28
constrained, time 3-41
constraint 12-6
view 11-21
context-sensitive menu A-6
contour line
displaying 18-9
spacing 18-10
thickness 18-10
Contour Lines page 18-9, 18-10
contrail 8-9
contrast 16-4
control, remote 1-15
control object 3-25
concepts 3-25
count of 6-23
name 3-25
selecting 6-4, 6-5
controller 3-17
component 3-17, 3-18
making decisions and performing tasks 3-17
controlling
messages B-14
remote 11-27
conventions, manual xxi
coordinate
observer 11-3
point 18-16
coordinate system 1-12
3D 11-6
changing unit of A-9
geocentric 3-35
supported 3-34
count, object 6-23
creating
inset view 8-8
creating (continued)
observer mode 10-3
selection group 6-8
selection group from menu 6-8
crepuscular ray 17-9
CTDB file 3-32
cube map, recalculating 17-12
cultural feature, information 6-15
curvature of the earth
display, mode 8-22

D

-d command-line option, vrfSim B-7
DAE file 3-32
daemon, starting back-end as B-7
data
directory 2-4
specifying location of B-3
elevation 3-32
port 3-17
synchronizing 5-24
database 3-35
See also terrain database
berkeley 18-17
ground-db 18-17
Kabul 18-18
large 3-36
LittlePond 18-18
loading 3-36
Local World Procedural Terrain 18-18
Makland 18-18
nellis 18-18
paging 3-37
SanLuisObispo 18-18
saving as GDB 5-21
terrain 1-12, 3-30
loading 5-19
TerraSim_SampleUrban 18-18
VR-TheWorldOnline 18-19
VR-Village 18-19
dead-reckoning 8-5
debug mode, running from vrfLauncher B-13
decal effects 8-9
enabling and disabling 8-9
models 8-11
decision making 3-17
decreasing speed of navigation 11-21
default
settings, restoring 5-22

default (continued)
view 11-25
Delete button 11-25
deleting
altitude point 18-5
extended label 7-12
intervisibility object 14-11
messages from object console 7-32
object from selection group 6-10
observer 10-8
mode 10-4
plug-in
configuration 5-37
XML file 5-38
selection group 6-11
view 11-25
demonstration, mode A-6
depth
bits B-3
focal 8-32
depth of field 8-32
configuring 8-34
enabling and disabling 8-32
deselecting
entity 6-7
object 6-7
designator 3-15
destination address B-6, B-10
detaching, observer from object 12-5
detonation
effects 8-14
event
mapping sound effects to 15-4
sound effects 15-2
line 8-14
result, sound mapping 15-8
sound mapping 15-8
Detonation Sound Editor page 15-8
Device address B-6, B-10
dialog box
Aggregate Information 6-15
Application Settings
GUI Layout page A-4
Mouse Mappings page 11-13
Performance Options page 8-6
Session Options page 5-7, 11-19
Audio Settings 15-2, 15-3, 15-5, 15-6, 15-7, 15-8
Connection Information 5-31
dialog box (continued)
Display Settings 5-22
Aggregate Display Settings page 8-15, 8-18, 9-7, 9-10, 9-11
Cockpit Display Settings page 8-26
Display Units Settings page A-11
Entity Display Settings page 7-5, 7-6, 7-14, 7-15, 7-19, 7-20, 7-32, 8-3, 8-4, 8-19, 8-35, 8-36, 10-9
Grid Lines Settings page 18-8
Intervisibility Settings page 14-5, 14-6, 14-8, 14-9, 14-10
Lighting Render Settings page 17-5, 17-10, 17-11
Observer Settings page 7-18, 7-19, 7-23, 7-27, 7-28, 8-9, 8-11, 8-33, 10-3, 10-4, 10-5, 10-8, 11-5, 11-20, 11-23, 17-4, 18-7
Ocean Render Settings page 17-6, 17-7, 17-12
Radio Communications Settings page 8-21, 8-22, 8-23
Range Rings Settings tab 7-24
Render Settings page 8-34, 17-8, 17-9, 17-11, 18-11
Sensor Settings page 16-4, 16-6
Shadow Settings page 8-29, 8-31, 17-14, 17-15
Symbol Decoration Settings page 7-7
Tactical Graphics Display Settings page 6-13, 6-14, 19-2
Terrain Profile Settings page 14-11
Track History Settings page 8-12
Entity Information 6-15, 7-26
Environmental Object Information 6-15
Experimental Features 11-18
License Setup 2-6
Plugins 5-32
Scenario Startup 5-5, 5-7
Simulation Connections Configuration 5-32
Stealth Information 6-15
Terrain Settings 18-12
Visible Surfaces page 18-2
diffuse, color 17-10
DI-Guy
character data file B-7
configuring, animations and appearances B-7
license 1-18
directory
appData B-7

directory (continued)
data 2-4
installed 2-4
DIS
command-line options B-18
Enumeration Document 3-10
enumerations document 4-9
exercise ID B-7, B-11
IPV6 B-11
port B-6, B-11
port number B-18
self reflection B-11
specifying broadcast modes B-18
starting in B-3
startup tutorial 5-8
timeout B-7
disabling
buoyancy 8-36
cockpit displays 8-25
decal effects 8-9
entity labels 7-6
environment 17-15
intervisibility 14-9
radio communication lines 8-21
range rings 7-21, 7-23
Scenario Startup dialog box 5-7
shadows 8-29, 17-13
speed scaling 11-21, 11-22
squawk indicators 8-21
tactical smoke 8-18
track histories for all entities 8-11
trailing effects 8-9
wireframe mode 18-10
disaggregated aggregate 3-20
disaggregation area 3-25
display
dragging 11-14
loading configuration B-3
navigating 11-3
unit, changing A-9
display engine 4-2
Display Engine Configuration Editor Panel 10-5
display mode, curvature of the earth 8-22
Display Settings dialog box 5-22
Aggregate Display Settings page 8-15, 8-18, 9-7, 9-10, 9-11
Cockpit Display Settings page 8-26
Display Units Settings page A-11
Display Settings dialog box (continued)
Entity Display Settings page 7-5, 7-6, 7-14, 7-15, 7-19, 7-20, 7-32, 8-3, 8-4, 8-19, 8-35, 8-36, 10-9
Grid Lines Settings page 18-8
Intervisibility Settings page 14-5, 14-6, 14-8, 14-9, 14-10
Lighting Render Settings page 17-5, 17-10, 17-11
Observer Settings page 7-18, 7-19, 7-23, 7-27, 7-28, 8-9, 8-11, 8-33, 10-3, 10-4, 10-5, 10-8, 11-5, 11-20, 11-23, 17-4, 18-7
Ocean Render Settings page 17-6, 17-7, 17-12
Radio Communications Settings page 8-21, 8-22, 8-23
Range Rings Settings tab 7-24
Render Settings page 8-34, 17-8, 17-9, 17-11, 18-11
Sensor Settings page 16-4, 16-6
Shadow Settings page 8-29, 8-31, 17-14, 17-15
Symbol Decoration Settings page 7-7
Tactical Graphics Display Settings page 6-13, 6-14, 19-2
Terrain Profile Settings page 14-11
Track History Settings page 8-12
Display Settings Toolbar 7-8
Display Units Settings page 6-14, 7-20, A-11
Displaying 17-2
displaying
contour lines 18-9
entity information 7-3
feature data 18-13
full screen A-6
grid line 18-7
height-above-terrain lines 6-14, 7-20
intervisibility
lines 14-9
object 14-6
label
2D 7-7
entity 7-4
laser designator 8-17
levels of aggregation in Echelon View 9-4
objects 6-11, 6-13
other MAK 3D GUI viewers 10-9
panel A-3
range rings 7-21
globally 7-23
per entity 7-23

displaying (continued)
resources 7-26
Scenario Startup dialog box 5-7
stealth models 10-9
tactical graphics 6-13
toolbars A-3
track histories 8-11
trailing effects 8-9
transient intervisibility 14-5
vertex labels 6-14
view 11-25
distance, specifying unit A-9
Distributed Interactive Simulation 1-23
See also DIS
DLL 5-32
plug-in, removing 5-36
specifying for plug-in 5-36
dockable panel A-2
docking, panel A-2
documentation, location of 2-4
dragging
terrain 11-13
view 11-14
dropping from exercise 3-8
DTED, file 3-32
dust cloud 8-9
dynamic, features B-8
dynamic lighting 17-3
dynamic linked library 5-32
dynamic ocean 1-14
wake and spray, disabling 8-35

E

-E command-line option B-3
earth file 1-19, 3-34
hiding layers 18-12
Earth Layers page 18-12
echelon ID 3-13, 3-14, 3-15
aggregate member 3-22
assignment of 3-16
echelon level, ghosted icons by 9-7
Echelon View 6-2, 9-4
expanding and collapsing aggregates 9-4
ghosted aggregates 9-6
levels of aggregation 9-4
ECW 3-33
editing
intervisibility object 14-7
observer mode 10-4
effect
fire and detonation 8-14
lighting 1-15
model 4-8
trailing 8-9
electromagnetic emissions 8-19
element definition, common icons A-7
elevation
coloring terrain by 18-4
data 3-32
intervisibility point, configuring 14-8
Elevation Color page 18-4, 18-5, 18-6
Embarkation View 6-2
embarked object, information 6-18
embedded window 4-2
emitter
information 6-18
systems 8-19
enabling
buoyancy 8-36
cockpit displays 8-25
decal effects 8-9
entity labels 7-6
intervisibility 14-9
radio communication lines 8-21
range rings 7-21, 7-23
Scenario Startup dialog box 5-7
shadows 8-29, 17-13
sound effects 15-2
speed scaling 11-21, 11-22
squawk indicator 8-21
tactical smoke 8-18
time management 5-17
in back-end 5-18, B-17
track histories for all entities 8-11
trailing effects 8-9
wireframe mode 18-10
engagement, information 6-20
entity
adding to, selection group 6-9
altitude, displaying 7-20
attaching observer to 12-2, 12-6
attribute, information 6-18
behavior 3-19
body frame coordinate system 11-6
bounding volume 7-27, 7-28
centering observer on 11-19
clamping to ground 8-3
communication model 3-16
concepts 3-13

entity (continued)
console message 7-28
count 6-23
display settings B-3
extents, zooming on scenario load 11-19
filtering, attachment list 12-4
ghosted 9-6
ground clamping 8-4
icon, changing size 7-14
identifiers 3-13
information 6-15
multiple 6-22
inset view 8-7, 8-8
instrument panel, configuring 8-28
intervisibility 14-3
label 3-14, 7-7
2D 7-4, 7-7
3D 7-3
color 7-3
enabling and disabling 7-6
extended 7-10
pinning to display 7-6
level body frame coordinate system 11-6
local and remote 3-10
mapping sound effects to 15-4
messages, viewing 7-30
model 4-8
name 7-3
length restrictions 3-14
notification level, object console 7-29
organization 3-15
pinning intervisibility object to 14-9
primary 12-9, 12-12
range ring 7-23
resource, monitoring 7-26
saving console messages 7-31
scaling, 3D model 7-16
secondary 12-9, 12-12
selecting 6-4, 6-5
from Object Console Summary Panel 6-6
in Objects List Panel 6-6
shadows 8-29, 17-13
sound effects 15-2
state 3-18
surface, buoyancy 8-36
task 3-19
track history 8-11
trailing effects 8-9
type 4-9
enumeration 7-3
entity (continued)
unselecting 6-7
views of 6-2
VR-Forces 3-10
wake, enabling and disabling 8-35
zooming, to extents 11-19
zooming to extents 11-18
Entity Display Settings page 7-5, 7-6, 7-14, 7-15, 7-19, 7-20, 7-32, 8-3, 8-4, 8-19, 8-35, 8-36, 10-9
Entity Editor 1-19
Entity Information dialog box 6-15, 7-26
Entity List, filtering 6-6
Entity Sound Editor page 15-5, 15-6, 15-7
entity type
mapping, loading B-3
mapping to element definition 4-9
Entity View 6-2
entity-level modeling 3-11, 3-20
EntityLevel.sms 3-11
enumeration, document 4-9
Enumeration and Bit Encoded Values for Use with Protocols for DIS Applications 3-10
environment
disabling 17-15
loading configuration B-3
environment variable
MAKLMGRD_LICENSE_FILE 2-8
RTI_CONFIG 2-10
Environmental Object Information dialog box 6-15
Environmentals List, filtering 6-6
Environmentals View 6-2
event 3-18
exaggerated reality 4-5
exaggerating, 3D models 7-16
executable, installation locations 2-4
execution name, HLA, specifying B-15
exercise, clock 3-39
exercise ID, DIS B-7, B-11
exiting
application 5-24
full screen window A-6
simulation engine 5-24
expanded label 7-3
pinning to display 7-6
expanding
aggregate 9-2, 9-3
all 9-3
Experimental Features dialog box 11-18

explosion 8-14
exporting
  settings 5-22
  terrain to GDB 5-21
extended label 3-26, 7-10
  adding 7-11
  deleting 7-12
  index, changing 7-12
extent, zooming to 11-18
external communication model 3-16
external communications effects server, configuring VR-Forces for 1-18
extracting
  prop, from terrain 3-33
extruded geometry 3-34
eyepoint. *See* observer

F

-F command-line option B-4, B-12
-f command-line option B-5, B-10, B-16
factory settings, restoring 5-22
fan, intervisibility 14-3
far clip, ocean planar reflection 17-7
favorite, observer view 11-25
FDD file 2-10
feature 1-13
  coordinate 18-16
  data, displaying 18-13
  dynamic B-8
  geometry 3-34
  layer, adding 3-33
  soil type 18-16
  vector, list of 18-14
Feature Settings page 18-15
FED file 2-10
  name B-5, B-10
  specifying B-17
federate
  name B-6, B-10
  time managed 3-41
federation execution, specifying name B-15
federation time 3-41
field of view 11-16, 11-17
file
  FED 2-10, B-17
  installed 2-4
  log 5-39
  RID 2-10, B-15
  rid.mtl B-15
file (continued)
  scenario 3-9
  transporter B-3
  vrfGui.log 5-39
  vrfSim.log 5-39
filtering
  earth file layers 18-12
  object attachment list 12-4
  object lists 6-6
fire
  effects 8-14
  event
    mapping sound effects to 15-4
    sound effects 15-2
  line 8-14
Fire Sound Editor page 15-8
fixed-frame 5-17
fixed-frame best-effort 3-40
fixed-frame run-to-complete 3-40
fixed-frame-run-to-complete 5-17
flaming effects 8-14
flat earth, coordinate system 11-6
FLEXlm 2-5
FLIR 16-2
floating, panel A-2
FLT file 3-32, 4-8
  mapping to models 4-9
focal depth 8-32
focal range 8-32
focus, scene 8-32
fog of war B-3
Follow mode 11-7, 12-6, 12-7
FOM 2-9
  module B-5, B-10
FOM Mapper
  initialization data B-5, B-10
  library name B-5, B-10
  specifying at startup B-16
footprint 8-9
force 3-15, 7-3
formation 3-23
  closing 3-23
frame rate B-9
  exercise clock 3-40
  log file B-8
  setting B-4
frame-mode parameter 5-18
  time management 5-17
free-fly mode 12-6

frequency
  radio communication line 8-23
  sampling for terrain profile 13-3
front-end 3-3
  coordinating multiple 3-7
  feature representation 3-34
  joining session at startup 5-16
  multiple 3-3
  process ID B-7
  sending messages 3-4
  starting 5-3
fuel, status 6-16
Full Color Mode 16-5
full screen
  mode B-3
  window 4-2, A-6

G

-G command-line option B-4, B-14
gain 16-2
  changing 11-23
  sensor effects 16-4
GDB terrain 3-32
  configuring soil type 18-2
  saving as 5-21
  shapefile 3-33
geocentric
  coordinate system 3-35, 11-6
  terrain 3-35
geometry, extruded 3-34
GeoTIFF 3-33
ghosted aggregate 9-6
ghosted icon, by echelon level 9-7
GIF 3-33
GL Studio 1-20, 8-25
  instrument panel 8-27, 8-28
  license 1-21
global plan, file 3-9
global setting 5-24
God ray 17-9
graphic
  aggregate combat 8-15
  depth B-3
  fire and detonation 8-14
  mode, wireframe 18-10
graphical object 3-25
  showing and hiding 6-11, 6-13
graphical user interface 3-3
grid line
  configuring 18-8
  displaying 18-7
Grid Lines Settings page 18-8
ground clamping 8-3
  configuring 8-4
Ground Clamping Cutoff Distance Scale 8-4
ground-db, database 18-17
group, selection 6-7
GUI 3-3
  localizing 2-11
  settings, file 3-9
GUI Layout page A-4
gumball 6-16

H

-H command-line option B-5, B-6
-h command-line option B-4, B-8, B-12
HarrisAfghanVillage, terrain 18-17
HAT, *See also* height-above-terrain line
heading 7-3, 11-3
  indicator 7-4
  intervisibility line 14-10
  observer 11-4
  rotating icon to 7-15
  specifying measurement unit A-9
headlight 17-3
height-above-terrain line 6-11, 7-20, 11-21
  displaying 6-14
help B-8, B-12
hiding
  3D models 7-19
  intervisibility lines 14-9
  objects 6-11, 6-13
  panel A-3
  symbol decorations 7-8
  toolbar A-3
  vertex labels 6-14
hierarchy
  displaying levels of aggregation 9-4
  entity 3-15
HLA
  command-line options B-15
  federation execution name, specifying B-15
  specifying federation execution B-15
  starting in B-4
  startup tutorial 5-8
  time management 3-41
host address B-5, B-6

hostility relationships file 3-9
hostname, license server 2-6

I

-I command-line option B-4
-i command-line option 5-8, B-5, B-6, B-8
icon
  2D, displaying label 7-7
  borders 7-13
  common for model definitions A-7
  heading, rotating to 7-15
  level of nesting 9-6
  location of 2-4
  notification 7-32
  scaling, 2D 7-14
  settings 5-22
  size, changing 7-14
ID
  echelon 3-14, 3-15
  object 3-14
  session 5-3, B-5, B-6
  site B-9
identifier, site and application 3-5
identifying, entities 3-13
IFF, information 6-18
image
  format, supported 3-33
imagery, filtering 18-12
importing, settings 5-22
increasing speed of navigation 11-21
independent mode, tutorial 5-8
independent setting, global and observer 5-24
independent task 3-28
index
  extended label, changing 7-12
indicator 7-6
  heading 7-4
  squawk 8-20
information
  connection 5-31
  entity label 7-3
  multiple objects 6-22
  object 6-15
infrared 16-2
inserting, altitude point 18-5
inset view 8-7, 8-8
installing, RTI 2-10
instancing B-4
instrument panel 8-27
  configuring 8-28
  GL Studio 8-27, 8-28
instrumentPanelCategory parameter 8-28
interaction 3-41
  signal 8-20
intermediate contour line 18-9
interval, contour line 18-10
intervisibility 3-38
  enabling and disabling 14-9
  entity 14-2, 14-3
  fan 14-2, 14-3
    moving 14-7
  grid 14-2
  line 3-25
    hiding 14-9
    information 14-10
  object
    configuring 14-8
    deleting 14-11
    displaying 14-6
    editing 14-7
    transient and permanent 14-4
  pinning object to entity 14-9
  point-to-point 14-2
  terrain profile 14-11
  terrain profile line 13-5
  types 14-2
Intervisibility Settings page 14-5, 14-6, 14-8, 14-9, 14-10
IPV6 B-7
  DIS B-11

J

joining exercise, late 3-8
JPEG 3-33
JPG 3-33
JRM Technologies 16-2
jumping, observer 11-15

K

Kabul, database 18-18
key mapping 11-10
keyboard
  attaching to objects from 12-3
  changing speed 11-23
  control A-6
  creating selection group from 6-9

keyboard (continued)
inset views 8-8
mappings 11-6, 11-9
movement functions 11-6
moving observer 11-9
keyword
largeCompass 11-5
smallCompass 11-5
Kynapse 1-17

L

-L command-line option B-4, B-8
-l command-line option B-4
label 3-13
2D
background 7-9
displaying 7-7
showing and hiding 7-8
entity 3-14, 7-7
2D 7-4
3D 7-3
enabling and disabling 7-6
extended 7-10
extended 3-26
object 3-14, 3-26, 6-14, 7-7
pinning to display 7-6
tactical graphics 3-26, 6-14
Lambert Conformal Conic, coordinate system 11-6
language
specifying B-4
for GUI B-14
largeCompass, keyword 11-5
laser designator, displaying 8-17
Last Clicked Location Panel 18-16
late joiner 3-8
latitude 7-3
Launcher, VR-Forces 5-4
layer
earth file, hiding 18-12
feature, adding 3-33
LCC, coordinate system 11-6
length
entity name 3-14
intervisibility object 14-7
track history 8-12
level, notification for messages B-14
level observer 11-6
level-of-detail 11-16, 11-17
library, dynamic 5-32
license
3D models 1-22
DI-Guy 1-18
GL Studio 1-21
OpenSceneGraph 1-22
osgEarth 1-23
SilverLining 1-20
SpeedTree 1-21
License Manager 2-5
license server, hostname 2-6
License Setup dialog box 2-6
lifetime
radio communication line 8-22
squawk indicator and communication line 8-22
lighting
advanced 17-4
disabling 17-15
dynamic 17-3
effects 1-14, 1-15
Lighting Render Settings page 17-5, 17-10, 17-11
line
creating, opening terrain profile 13-3
editing, opening terrain profile 13-3
fire 8-14
height-above-terrain 6-11, 7-20
intervisibility 3-25
information 14-10
radio communication 8-20
supply 8-18
terrain profile 3-25, 13-5
for intervisibility 14-11
vertex, showing in terrain profile 13-3
linear feature, list of 18-14
linear scaling 7-16
line-of-sight 3-38
communication line 8-22
terrain profile 14-11
list, vector feature 18-14
LittlePond terrain database 18-18
loading
database 3-36
display configuration B-3
entity display settings B-3
entity type mapping B-3
environment configuration B-3
plug-ins 5-32, 5-33, 5-34, B-3, B-4, B-8, B-15
preventing B-5
saved views 11-26
scenario B-5, B-8

loading (continued)
settings 5-22
terrain 5-19
from command line B-5
paged B-8
local
entity 3-10
object 3-10
Local World Procedural Terrain database 18-18
locale, specifying B-4
localization, specifying B-4
localizing, GUI 2-11
location
compass 11-5
jumping to 11-15
observer 11-3
LOD 11-16, 11-17
paged, scale factor 11-18
scale, setting 17-7
LOD distance, buoyancy 8-36
log file 5-39, B-8, B-9
frame rate statistics B-8
messages sent to B-14
sending entity messages 7-31
specifying B-4
vrfSim B-8
longitude 7-3
LWO file 3-32

M

MÄK RTI, configuring 2-10
making decisions 3-17
Makland, database 18-18
MAKLMGRD_LICENSE_FILE, environment variable 2-8
managing, plug-ins 5-32
manual
conventions xxi
location of 2-4
map
cube, recalculating 17-12
texture 1-15
Map Scale Toolbar 18-17
mapping
keyboard 11-10
model
to entity type 4-9
to remote observer 10-9
movement to keyboard 11-9
mapping (continued)
sound effects 15-4
sound file to entity 15-5
marking text 7-3
material, ambient value 17-10
MEDF, file 3-32
menu
context-sensitive A-6
Settings 5-22
tearing off A-5
message B-8
console B-8
entity 7-28
saving 7-31
notification level B-4
notification of 7-32
object, viewing 7-30
sending, between front-end and back-end 3-4
sent 3-18
session, configuring 5-16
setting levels B-14
state update 10-8
time stamp ordered, TSO 3-41
view control 11-27
middle-click menu A-6
MIL-STD 2525B color 7-19
aggregate 9-11
Mimic mode 11-7, 12-6, 12-8
Mimic Track mode 12-9
mode
Absolute 12-6
Follow 12-7
Mimic 12-8
Mimic Track 12-9
observer 4-3
Space Follow 12-10
specifying B-18
speed scaling 11-22
Tether 12-11
Tether Track 12-12
Track 12-13
view 11-7, 12-6
model 4-8
3D
exaggerating entity 7-16
hiding 7-19
scaling 7-19
decal effects 8-11
instancing B-4
layer, filtering 18-12

model (continued)
mapping
common icons A-7
to FLT files 4-9
to remote observer 10-9
name 7-3
physical 3-18
set, changing 7-15
stealth 10-9
trailing effects 8-11
types 4-8
vehicle 1-6
model definition 4-9
compass 11-5
model set 7-15
modeling
aggregate-level 3-11, 3-20
entity-level 3-11, 3-20
monitoring, entity resources 7-26
mouse
changing speed 11-23
drag terrain 11-13
drag view 11-14
moving observer 11-13, 11-14, 11-16
orbiting terrain 11-14, 11-16
teleporting observer 11-15
Mouse Mappings page 11-13, 11-16
movement
aggregate 3-23
speed scaling 11-21
moving
camera 11-12
intervisibility fan 14-7
observer 11-3, 11-11
mouse with 11-13, 11-14, 11-16
options 11-11
MTD file 3-32
MTL file 3-32
MTP file 3-32
multicast mode, specifying B-18
multicast time-to-live B-18
multiple
back-ends 3-3, 3-8
front-ends 3-3
coordinating 3-7
object, information 6-22
VR-Forces processes 3-3
Multiple Object Information window 6-22
munition, display of path and detonation 8-14
mutually exclusive task 3-28
muzzle flashes 8-14

N

-N command-line option B-5, B-10
-n command-line option B-4, B-8, B-14
name 3-13
control object 3-25
displaying 7-4
entity 3-14, 7-3
navigating, view 11-3
navigation
attached context 11-7
options to 11-11
speed scaling 11-21
nellis, database 18-18
network
connecting to, automatically B-3
new, window 5-13
night vision 16-2
no dialog box B-12
noise 16-4
normal map 1-15
notification, icon 7-32
notification level B-8
setting B-14
setting for entity console 7-29
specifying B-4
NVG 16-2

O

-O command-line option B-4
OBJ 4-8
file 3-32
object
adding, to selection group 6-8, 6-9
attaching observer to 12-2
concepts 3-10
control 3-25
count 6-23
detaching observer from 12-5
exaggerating models 7-16
filtering list of 6-6
ID 3-13, 3-14
intervisibility 14-4
pinning to entity 14-9
label 3-14, 3-26, 6-14, 7-7
list, using 12-2
local 3-10

object (continued)
messages, viewing 7-30
model 4-8
multiple, information 6-22
orbiting 11-12
overlay 3-25
remote 3-10
selecting 6-4
in Objects List Panel 6-6
showing and hiding 6-13
unselecting 6-7
views of 6-2
VR-Forces 3-10
object console 7-28
clearing 7-32
notification of messages 7-32
setting notify level 7-29
Object Console Summary Panel 6-4, 7-30, 11-19
answering questions 7-31
selecting entity 6-6
object map file 3-9
missing back-ends 3-9
object parameter database 3-10
objectConsoleNotifyLevel parameter 7-29
Objects Console Summary Panel 6-21
Objects List Panel 6-2, 6-4, 6-6, 9-4
selecting object in 6-6
observer 11-3
adding 10-5, 10-6
attaching
articulated part 12-4
to entities 12-5
to entity or prop 12-2
attachment
in 2D 12-14
order 12-3
camera 11-12
centering on entity 11-19
compass 11-4
coordinate system
2D 11-7
3D 11-6
detaching from object 12-5
dragging
terrain 11-13
view 11-14
heading 11-4
inset view, in 8-8
location and orientation 11-3
observer (continued)
mode 4-3
changing 10-2
creating 10-3
deleting 10-4
editing 10-4
renaming 10-4
setting 5-24
switching projection 4-3
XR 4-5
movement 12-6
functions 11-6
speed 11-23
moving
keyboard with 11-9
mouse with 11-13, 11-16
orbiting 11-12
orientation 11-14
range for sound effects 15-3
removing 10-8
saving view 11-23
teleporting 11-15
view
attachment change 12-3
transitioning between 11-25
ways to move 11-11
zooming 11-16
2D 11-18
3D 11-17
entity extents 11-19
zooming to extents 11-18
Observer Control Panel 11-21
Observer Information Panel 11-3
Observer Settings page 7-18, 7-19, 7-23, 7-27, 7-28, 8-9, 8-11, 8-33, 10-3, 10-4, 10-5, 10-8, 11-5, 11-20, 11-23, 16-4, 17-4, 18-7
Observer Settings Toolbar 7-15, 7-19
Observer Views Panel 11-23
loading saved views 11-26
saving views 11-26
obstacle 3-25, 3-28
obstructed terrain, viewing in terrain profile 14-11
ocean
planar reflection 17-6
configuring 17-7
planar reflections, performance 17-7
synchronized B-4
ocean planar reflection
texture height 17-7
texture width 17-7

Ocean Render Settings page 17-6, 17-7, 17-12
opacity, 2D label background 7-9
OPD Editor 1-19
OpenAL 15-3
OpenFlight 4-8
  file 3-32
  format 1-6
opening
  inset view 8-8
  terrain 3-36
OpenSceneGraph 1-3, 1-20
  license 1-22
option
  command-line B-3
  navigation 11-11
orbit, observer 11-12
orbiting
  terrain, with mouse 11-14, 11-16
order of battle 3-9
organization
  aggregate 3-22
  entity of 3-15
orientation 11-3
  clamping 8-4
  displaying 7-4
  icon 7-15
  observer 11-3, 11-13, 11-14, 11-16, 12-6
  specifying measurement unit A-9
OSG file 3-32
osgEarth 1-20
  license 1-23
osgearth_cache 1-19
Out-the-Window, observer mode 5-24
overlay 1-13
  concepts 3-25
  hiding graphics on 6-13
  object 3-25
    selecting 6-4
Overlays tab 6-13
Overlays view 6-2

P

-P command-line option B-6, B-11, B-18
-p command-line option B-6, B-10
page
  Audio Settings 15-2, 15-3
  Detonation Sound Editor 15-8
  Display Units Settings 6-14, 7-20
  Earth Layers 18-12
page (continued)
  Elevation Color 18-4, 18-5, 18-6
  Entity Sound Editor 15-5, 15-6, 15-7
  Feature Settings 18-15
  Fire Sound Editor 15-8
  Mouse Mappings 11-16
  Observer Settings 10-5, 16-4
  Render Settings 8-34, 18-4
  Resource Information 7-26
  Sensor Settings 16-4
  Session Options 7-11, 7-12, A-8
  Shadow Settings 8-29, 8-31, 17-14, 17-15
  Symbol Decoration Settings 7-9
Paged LOD Magnification Scale Factor 11-18
paged terrain, loading B-8
paging, terrain 3-37
panel
  displaying and hiding A-3
  dockable A-2
  docking A-2
  floating A-2
  Last Clicked Location 18-16
  Object Console Summary 7-30
  undocking A-2
parallel processing B-9
parameter
  assertOnBlockingTerrainCalls 3-37
  frame-mode 5-18
    time management 5-17
  instrumentPanelCategory 8-28
  objectConsoleNotifyLevel 7-29
  providing information to components 3-17
  RTI_timeMgmt 5-18
  RTI_useRtiExec 5-18
  runInTimeManagementMode 5-18
patch, terrain 3-32
path, specifying for RTI 2-10
PDU
  sending 10-8
  view control 11-27
performance 11-17
  anti-aliasing B-3
  asynchronous I/O B-18
  attached observer 12-5
  cube map calculation 17-12
  frame rate B-9
  graphics, depth B-3
  HAT lines 6-14, 7-20
  instancing B-4
  main thread B-9

performance (continued)
ocean planar reflections 17-7
shadows 8-31, 17-15
stencil bits B-5
Performance Options page 8-6
permanent intervisibility object 14-4
personnel, status 6-16
phase line 3-25, 3-27
physical model 3-18
pitch 11-3
plan
abandoning 3-28
concepts 3-29
file 3-9
plan view 4-5
Plan View window 5-13
planar reflection 17-6
plug-in
adding 5-35
configuration, deleting 5-37
DLL
removing 5-36
specifying 5-36
list, viewing 5-38
loading 5-32, 5-33, 5-34, B-3, B-4, B-8, B-15
managing 5-32
prevent loading B-5
version, checking B-3
XML file 5-35
deleting 5-38
Plugins dialog box 5-32
Plugins Editor 5-33, 5-35, 5-36, 5-37, 5-38
PNG 3-33
point 3-26
feature, list of 18-14
orbiting around 11-12
terrain profile, sampling rate 13-3
point feature, in front-end and back-end 3-34
point-to-point communications, specifying B-18
point-to-point intervisibility 14-2
POL, status 6-16
polygon 3-27, 3-36
not going through 11-21
popup menu A-6
port 3-17
default B-18
group 3-17
number, specifying at startup B-18
specifying, at startup B-18
position 11-3
observer 11-13, 11-16, 12-6
preprocess call B-9
Presagis 1-22
primary attached entity 12-9, 12-12
primary color, aggregate 9-11
printing, window 5-13
process, VR-Forces 3-3
process ID, front-end B-7
product
technical support xx
upgrades xx
projected terrain database 11-6
projection
supported 1-13
switching 2D to 3D 4-3
prop 3-34
attaching observer to 12-2, 12-5
configuring 3-33
extracting, from terrain 3-33
filtering attachment list 12-4

Q

-q command-line option B-9
Qt Linguist 2-11
question, scripted task 7-31

R

-r command-line option B-9
radio
communication line 8-20
color 8-23
enabling and disabling 8-21
lifetime 8-22
Radio Communications Settings page 8-21, 8-22, 8-23
range
focal 8-32
ground clamping 8-4
intervisibility line 14-10
range ring 7-21
configuring display of 7-24
displaying for entity 7-23
enabling and disabling 7-23
sensor 7-21
Range Rings Settings tab 7-24
ray
crepuscular 17-9

ray (continued)
intervisibility 14-3
sun 17-9
Real DB 1-22
real time, running faster or slower than 3-40
recalling
view 11-25
view state 11-23
receive buffer size B-6, B-11
reflection
map 1-15
planar 17-6
regulating, time 3-41
rejoining exercise 3-8
remapping, back-ends 3-9
remote
control 1-15
of the observer 11-27
entity 3-10
object 3-10
stealth, displaying model 10-9
removing
objects from selection group 6-10
observer 10-8
view 11-25
renaming
observer mode 10-4
selection group 6-10
Render Settings page 8-34, 17-8, 17-9, 17-11, 18-4, 18-11
reorganizing
aggregate, versus closing formation 3-23
concepts for aggregates 3-22
replacing, saved view list 11-26
replicating, settings and data 5-24
resizeable window 4-2
resource
information 6-18
monitoring entity 7-26
Resource Information, page 7-26
restoring, view 11-25
result
detonation, sound mapping 15-8
revision, RPR FOM B-6, B-10
revolving, observer 11-12
RGB 3-33
RID file 2-10, B-15
rid.mtl file 2-10, B-15
right-click menu A-6
ring, range 7-21
roll 11-3
rotation path, smoothing 8-5
route 3-25, 3-27
altitude, displaying 6-14
RPR FOM 4-9, B-16
revision B-6, B-10
version B-6, B-10
RTI 2-9
configuring for time management 5-18
definition of 2-9
RTI_CONFIG environment variable 2-10
RTI_timeMgmt parameter 5-18
RTI_useRtiExec parameter 5-18
rtiexec 5-18
run mode, starting back-end in B-9
runInTimeManagementMode parameter 5-18

S

-S command-line option 12-5, B-5, B-6, B-10, B-11, B-18
-s command-line option B-9, B-13
sampling rate
intervisibility, configuring 14-8
SanLuisObispo terrain database 18-18
saved views file 3-9
saving
scenario 3-8
settings 5-22, 5-23
terrain as GDB 5-21
view 11-23, 11-24
to file 11-26
scale
factor, material ambient and diffuse value 17-10
terrain 18-17
Scale Factor slider 7-19
Scaled Speed check box 11-22
scaling
3D models 7-16
icon, 2D 7-14
model
3D 7-19
linear and XR 7-16
scenario 3-9
file 3-8, 3-9
loading B-5, B-8
at startup B-8
saving 3-8
startup observer view 11-25
zoom to extents on load 11-19

Scenario Merge 1-19
Scenario Startup dialog box 5-5, 5-7
scripted task, question 7-31
secondary
  attached entity 12-9, 12-12
  window 5-13
sector, aggregate 7-21
selecting
  aggregate 9-2
  entities 6-5
  enttiy, from Object Console Summary Panel 6-6
  object 6-4
    in Objects List Panel 6-6
  observer 10-5
  selection group 6-9
selection box 6-5
selection group
  adding object to new 6-8
  adding to 6-9
  creating 6-8
  creating from keyboard 6-9
  creating from menu 6-8
  deleting 6-11
  introduction 6-7
  removing object from 6-10
  renaming 6-10
  selecting 6-9
Selection Group View 6-2
Selection Groups Panel 6-8, 6-10, 6-11
self reflection, DIS B-11
send buffer size B-6, B-11
sending, messages 3-18
SenSim 16-2
sensor 3-17
  information 6-18
  mode, full color 16-5
  range ring 7-21
sensor effects 16-2, 16-3
  blur, noise, and gain 16-4
  configuring 16-4
Sensor Settings page 16-4, 16-6
server, terrain 3-36, 3-38
session 3-6
  ID 5-3, B-5, B-6
    assigning 5-8
    setting at startup B-8
  joining at startup 5-16
  message, configuring 5-16
Session Options page 5-7, 7-11, 7-12, 11-19, A-8
Set Appearance 17-3
set data request 3-30
set statement 3-30
  See also. set data request
setting
  color, aggregate bounding volume 9-11
  factory and default 5-22
  frame rate B-4
  icons 5-22
  loading 5-22
  LOD scale 17-7
  notification level B-14
    entity 7-29
  saving 5-22
  user 5-23
settings
  sound volume 15-3
  synchronizing 5-24
Settings menu 5-22
shader 1-15
shading, advanced 17-4
shadow
  configuring 8-31, 17-15
  enabling and disabling 8-29, 17-13
  text, 2D label 7-9
Shadow Settings page 8-29, 8-31, 17-14, 17-15
shape terrain data 3-32
Show Database Correlation Warning Dialogs 5-17
Show Session Terrain Change Prompts 5-17
showing
  objects 6-11, 6-13
  symbol decoration 7-8
  vertex labels 6-14
SHP, file 3-32
signal, interaction 8-20
SilverLining 1-20
  license 1-20
simple-radio-comm-model 3-16
Simthetiq 1-22
simulated world, navigating 11-3
simulation
  engine 3-3
  time 3-39, 3-41
Simulation Connections Configuration dialog box 5-32
Simulation Control toolbar 5-11
simulation engine
  closing 5-24
  exiting 5-24

simulation model set
entity-level and aggregate-level 3-11
specifying on command-line B-9
Simulation Selection toolbar 5-11
Simulation Time Scale, toolbar 5-11
Simulation Time toolbar 5-11
site ID 3-5, B-6, B-9
specifying B-9
at startup B-13
situational awareness 4-5
skybox cube map 17-12
omitting clouds 17-12
Skybox Cube Map Regeneration Threshold 17-12
smallCompass, keyword 11-5
smoke
plume 8-14
trail 8-9
smooth period 8-5
smoothing 8-5
SO 5-32
soil type
configuring for GDB 18-2
point and feature 18-16
sonar, information 6-18
sound effects 15-2
adjusting volume 15-3
enabling 15-2
mapping, weapon fire and detonation 15-8
mapping to entities 15-4
range 15-3
sound file, mapping to entity 15-5
sound mapping
channel 15-8
detonation result 15-8
Space Follow mode 12-10
spacing, contour line 18-10
specifying
application number B-7
FED file B-17
HLA federation execution B-15
language B-4
log file B-4
notification level B-4
point-to-point and multicast modes B-18
port, at startup B-18
simulation model set B-9
site ID B-9
terrain profile sampling frequency 13-3
specular, color 17-10
specular map 1-15
speed 7-3
gain 11-23
scaling 11-21
enabling and disabling 11-22
specifying measurement unit A-9
speed scaling, mode 11-22
SpeedTree 1-20
license 1-21
spot report B-3
spotlight 17-3
spray, enabling and disabling 8-35
squawk indicator 8-20
color 8-23
enabling and disabling 8-21
lifetime 8-22
starting
back-end 5-3
front-end 5-3
VR-Forces 5-3
startup
option B-1
tutorial 5-8
state
entity, information 6-18
update, sending 10-8
status
entity 7-26
personnel, ammunition, weapons, and POL 6-16
Stealth
observer mode 5-24
window 5-13
Stealth Information dialog box 6-15
stencil bits B-5
streaming, terrain 3-38
street light 17-3
subnet mask B-6
subordinate, information 6-20
sun, ray 17-9
sunbeam 17-9
supplies, information 6-20
supply line 8-18
support, technical xx
surface, clamping to 8-3
surface entity
buoyancy 8-36
wake 8-9
swell 1-14
symbol
decoration, showing and hiding 7-8

symbol (continued)
location of 2-4
Symbol Decoration Settings page 7-7, 7-9
synchronized, ocean B-4
synchronizing
back-ends 3-41
data and settings 5-24

T

-T command-line option B-9
tactical graphic 3-25
console, notification level 7-29
HAT lines 6-14
hiding from Overlays tab 6-13
information 6-15
label 6-14
labels 3-26
selecting 6-5
showing and hiding 6-11, 6-13
Tactical Graphics Display Settings page 6-13, 6-14, 19-2
tactical smoke, enabling or disabling 8-18
taillight 17-3
task
aggregate concepts 3-24
concepts 3-28
concurrent 3-28
entity 3-19
independent 3-28
mutually exclusive 3-28
performing 3-17
status 6-18
TDB Tool 3-30
tearing off menu A-5
technical support xx
teleporting observer 11-15
terminating, application 5-24
terrain
agility 1-12, 3-31
altitude point
changing color 18-6
deleting 18-5
inserting 18-5
analyzing 13-5
caching earth files 1-19
coloring by elevation 18-4
composability 3-31
coordinates 18-16
database 1-12, 3-30
terrain (continued)
dragging with mouse 11-13
extracting props from 3-33
formats, supported 3-32
GDB, soil type 18-2
geocentric 3-35
HarrisAfghanVillage 18-17
large 3-36
loading, from command line B-5
opening 3-36
orbiting, with mouse 11-14, 11-16
paged, loading B-8
page-in area 3-25
paging 3-37
patch 3-32
profile. See terrain profile
scale 18-17
server. See terrain server
shape 3-32
soil type 18-16
streaming 3-38
wireframe mode 18-10
terrain database 3-35
loading 5-19
at startup B-9
provided with application 18-17
saving as GDB 5-21
terrain profile
concepts 13-2
configuring 13-3
line 3-25, 13-5
opening automatically 13-3
sampling frequency 13-3
terrain profile line 13-5
track history 8-13
vertex, showing 13-3
Terrain Profile Settings page 14-11
Terrain Profile window 13-2, 13-3
intervisibility line 14-11
terrain. *See also* terrain database
terrain server 3-36, 3-38
Terrain Settings dialog box 18-12
Visible Surfaces page 18-2
Terrapage file 3-32
TerraSim 1-22
TerraSim_SampleUrban, database 18-18
Terrex 1-22
Tether mode 11-7, 11-19, 12-6, 12-11
Tether Track mode 12-12

text
in title bar A-8
shadow 7-9
texture, map 1-15
texture height, ocean planar reflection 17-7
texture width, ocean planar reflection 17-7
thickness, contour line 18-10
threat ring 7-21
TIFF 3-33
time, simulation 3-39
time constrained 3-41
time management
concepts 3-41
configuring 5-17
enabling 5-17
for back-end 5-18, B-17
RTI requirements 5-18
time of day 3-39
time regulating 3-41
time stamp B-5, B-9
time stamp ordered message 3-41
timeout, DIS B-7
timestamp B-5, B-9
tire track 8-9
title bar
text, adding A-8
To Echelon Level 9-7
toolbar
Attachments 12-2
button 5-11
Display Settings 7-8
displaying and hiding A-3
Map Scale 18-17
Objects 6-6, 9-4
Observer Settings 7-15, 7-19
track history
configuring 8-12
displaying 8-11
enabling and disabling for all entities 8-11
length 8-12
terrain profile 8-13
Track History Settings page 8-12
Track mode 11-7, 12-6, 12-13
tracking
entity, resources 7-26
trail, model 8-11
trailing effects 8-9
enabling and disabling 8-9
model 8-11
trajectory, smoothing 8-5
trajectory history. *See* track history
transient intervisibility
displaying 14-5
object 14-4
translating, GUI text 2-11
tread track 8-9
triangle, vertex label 6-14
turret, attaching to 12-4
tutorial, starting VR-Forces 5-8
TXP file 3-32

U

unconstrained 3-41
undocking, panel A-2
unit
display A-9
measurement A-9
Universal Transverse Mercator 3-35
unselecting
entity 6-7
object 6-7
update
messages 10-8
rate, intervisibility 14-8
upgrades xx
user
data, specifying directory B-5
settings 5-23
UTM 3-35
coordinate system 11-6

V

-v command-line option B-5, B-9, B-12
vapor trail 8-9
variable-frame run-to-complete 3-40
vector feature, list 18-14
vehicle, model 1-6
velocity
displaying 7-4
specifying measurement unit A-9
version
displaying B-5
plug-in, checking B-3
RPR FOM B-6, B-10
vertex
altitude, displaying 6-14
line, terrain profile 13-3
showing and hiding 6-14

view
clearing list 11-25
constraint 11-21
default 11-25
deleting 11-25
dragging with mouse 11-14
floor 11-21
inset 8-7, 8-8
loading saved 11-26
mode, Tether Track 12-12
navigating 11-3
observer, transitioning between 11-25
recalling 11-23
restoring 11-25
saved with scenario 3-9
saving 11-23, 11-24, 11-26
zooming 11-16
2D 11-18
3D 11-17
view control message 11-27
enabling processing of message 11-27
script 11-27
viewing
obstructed terrain in terrain profile 14-11
other MAK 3D GUI viewers 10-9
plug-in, data 5-38
visible surface, configuring 18-2
Visible Surfaces page 18-2
volume
sound, adjusting 15-3
vrfGui 3-3
vrfGui.log file 5-39
vrfLauncher
combined mode B-12
special command-line arguments B-13
vrfMsgDump 1-19
VR-Forces
entity or object 3-10
Launcher 5-4
starting 5-3
window 5-11, 5-19
vrfSim, log file B-8
vrfSim 3-3
daemon, starting as B-7
vrfSim.log file 5-39
VR-TheWorldOnline, database 18-19
VR-Vantage Control Toolkit 11-27
VR-Village, database 18-19

W

wake 1-14, 8-9
enabling and disabling 8-35
wall, not walking through 11-21
wall-clock time 5-17
waypoint 3-25, 3-26
altitude, displaying 6-14
weapon
fire, sound mapping 15-8
status 6-16
wind 1-14
window
Multiple Object Information 6-22
new 5-13
observer 10-5
printing 5-13
secondary 5-13
selecting entities in 6-5
specifying B-3
Terrain Profile 13-2
types 4-2
Windows, installing on 2-2
wireframe mode 18-10
world coordinates 11-3

X-Y-Z

-x command-line option B-6, B-7, B-10, B-11, B-15, B-19
XML, RTI configuration file 2-10
XML file
plug-in 5-35
deleting 5-38
XR mode 4-5
XR scaling 7-16
yaw 11-3
Z-fighting 12-5
zoom
scale factor, LOD 11-18
scenario, on load 11-19
switching from 3D to 2D 4-3
to entity extents 11-19
to extents 11-18
zooming
observer 11-16
2D 11-18
3D 11-17

VR-Forces Quick Reference Card

vrfSim Command-Line Options

Option	Description
(-a | --appNumber) *application_number*	Specifies the application number.
--additionalSystemScriptsDirectory *directory*	Specifies a directory where VR-Forces will look for system scripts when it populates the Scripted Task dialog box script list.
--appDataDir *directory*	Specifies the location of the appData directory.
--appIdRange *lower-upper*	Specifies a range of application IDs that represent objects generated by VR-Forces.
(-B | --batchScenarioFileName) *batch_file*	Runs the specified batch file. (For use with one back-end only.)
(-c | --frontEndPID) *PID*	The process ID for the front-end when running in combined mode.
--countryCodesMappingFile *file_name*	Specifies an alternative file that maps country codes to country names and DIS enumeration values.
(-d | --settingsFile) *file_name*	Specifies a configuration file to load instead of *vrfSimSettings.xml*.
--daemon	Starts that back-end as a daemon process (Linux only).
--diGuyAnimationsFile *file_name*	Specifies an additional configuration file for DI-Guy animations.
--diGuyCharacterDataFile *file_name*	Specifies an additional configuration file for DI-Guy character data.
--doNotLoadVrfPlugins	Do not load any VR-Forces plug-ins from the *./plugins* directory.
--dynamicFeaturesDisabled	Enables the dynamic features system.
--frameRateStatisticsLogFile *file*	Specifies the log file for frame rate statistics.
--fullyLoadTerrain	For paged terrains, load all pages.
(-h | --help)	Displays help information in a console window.
--hostAddressString *address*	The IP address of the host computer. (Device Address in Launcher.)
(-i | --sessionId) *session_ID*	Specifies a session ID.
(-L | --scenarioFileName) "*scenario_pathname*"	Loads the specified scenario. Mutually exclusive with the -T option.
--loadPlugin *plugin_file*	Specifies one or more plug-ins to load. (Accepted multiple times.)
--logFrameRateStatistics	Specifies that the back-end should log frame rate statistics.
--msdlSIDCMappingFile *file_name*	Specifies an alternative file for mapping SIDC codes to DIS enumerations.
(-n | --notifyLevel) *notification_level*	Sets the level for messages written to the console or the VR-Forces log file (*vrf.log*).
--outputLogFile *file_name*	The file to use to write out application output.
(-q | --doNotUseConsole)	Specifies that all *vrfSim* output go to the log file.
(-r | --startInRunMode)	Starts the back-end in run mode..
(-s | --siteId) *site_id*	Specifies the site ID.
--setMainThreadToHighPriority	If enabled sets the VR-Forces main thread to high priority. Windows only.
--simulationModelSet *sms*	Specifies the simulation model set.
(-T | --terrainDatabase) *terrain_database*	Specifies a terrain database to load in the back-end. Mutually exclusive with the -L option.

Copyright © 2015 VT MÄK. All rights reserved. Document ID: VRF-4.3.1-QRC-150625

vrfSim Command-Line Options

Option	Description
`--targetFrameRate` *rate*	Specifies the frame rate above which vrfSim will yield CPU time.
`--useAbsoluteTimeStamps`	Use absolute time stamps instead of relative time stamps.
`--disableParallelPreprocess`	Disables parallel processing for components that use it.
(`-v` | `--version`) *version*	Displays version information.
(`--` | `--ignore_rest`)]	Ignore all command-line options that are listed after this argument.
HLA Only	
(`-f` | `--fomMapperLib`) *libname*	FOM Mapper library name.
(`-F` | `--fedFileName`) *fedfile_name*	FED file name.
`--fomMapperInitData` *data*	FOM Mapper initialization data.
`--fomModules` *module*	Specifies an HLA Evolved FOM module. Can be used multiple times.
`--ignoreAdvisories`	Enables or disables HLA advisory messages. Default: true.
`--mimModule`	Specifies an HLA Evolved MIM module.
(`-N` | `--federateName`) *federate_name*	Specifies a unique name that can be assigned to the federate. Default: VR-Forces. (HLA Evolved only)
(`-p` | `--federateType`) *type*	Distinguish different categories of federates.
`--rprFomRevision`	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, or 2.0017)
(`-S` | `--localSettingsDesignator`) *designator*	Specifies a string that is passed to the RTI during federate connect when using HLA Evolved.
`--timeManagement`	Enables time management for the back-end.
(`-x` | `--execName`) *exec_name*	Execution name. Default: VR-Link.
DIS Only	
(`-A` | `--destAddrString`) *address*	Destination address.
`--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: 5.
`--mcastTtl`	Specifies the multicast time-to-live. Default: -1.
(`-S` | `--multicastAddresses`) *address1* ... *addressn*	Specifies one or more multicast addresses.
(`-P` | `--disPort`) *port*	DIS port.
`--recvBufferSize` *size*	Receive buffer size.
`--sendBufferSize` *size*	Send buffer size. Default: -1.
`--subnetMask` *mask*	Specifies the subnet mask.
`--suppressSelfReflect`	Represses self reflection of the DIS exercise connection.
`--useAsyncIO`	Use asynchronous I/O. Default: False.
`--useIpv6`	Specifies that the DIS exercise connection use IPV6.
(`-x` | `--exerciseId`) *ID*	Exercise ID.

vrfGui Command-Line Options

Option	Description
(-- | --ignore_rest)	Ignore all command-line options after this one.
--antiAliasing *level*	Set the anti-aliasing level (0, 2, 4, 8, or 16. Default: 5.
--appDataDir *directory*	Specifies the location of application data.
(-c | --showConsole)	Display a console window.
(-C | --autoConnect) {*filename* | *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. Default: 24.
--dis	Use the DIS protocol and enable DIS-specific options.
--dispSetting *filename*	Specifies the display configuration to load.
--doNotCheckPluginVersions	Do not check the version in a plug-in to make sure it can load in VR-Forces.
--doNotLoadVrfPlugins	Prevents loading of plug-ins in the *./plugins* directory.
--entDispSetting *filename*	Specifies the entity display setting to load.
--envSetting *filename*	Specifies the environment configuration file to load.
(-E | --entTypeMap) *filename*	Load the specified entity type mapping file. The file extension determines the type of mapping.
--fileTransporterReceivePort *port*	Port for the file transporter.
--fowPerspective *force*	Shows the ground truth and spot reports perspective for the specified force.
--fullScreen	Start in full screen mode.
--frameRate *rate*	Specifies a fixed frame rate for rendering. If 0, it is variable.
(-G | --locale) *language*	Specifies the locale (for localization).
--gui *filename*	Specify GUI configuration file to load.
(-h | --help)	Displays command-line usage information.
--hla13	Use HLA 1.3 and enable HLA-specific options.
--hla1516	Use HLA 1516 protocol and enable HLA-specific options.
--hla1516e	Use HLA Evolved protocol and enable HLA-specific options.
(-I | --unbatchedRendering)	Disables instance rendering.
(-l | --logFileName) *filename*	Specifies the log file name.
--loadPlugin *filename*	Specifies plug-ins to load. Accepted multiple times.
(-L | --loadObservers *filename1 filenamen*	Loads the saved observers in the specified files.
(-n | --notifyLevel) *notify_level*	Specifies the notification level for application messages. Range: 0-4.
(-O | --hasNTPSync)	Enables synchronized ocean between different display engines. System clocks must be synchronized.
--plugin *filename*	Load the specified plug-in. Can be used multiple times.
--pvd	Start the VR-Forces front-end with only 2D modes available.
(-S | --suppressHatIntersectOnAttach)	Suppress the periodic HAT intersection when attached.

Copyright © 2015 VT MÄK. All rights reserved. Document ID: VRF-4.3.1-QRC-150625

vrfGui Command-Line Options

Option	Description
`--scenarioFile` *filename*	Specifies a scenario to load.
`--stealth`	Start the VR-Forces front-end with only 3D modes available.
`--stencilBits` *bits*	Sets the number of stencil bits (0 or 8). Default: 0.
(*terrain_MTF_file* | *scene_MSF_file*)	Specifies a terrain (MTF) or scene (MSF) file to load.
`--unplug` *filename*	Do not load the specified plug-in. Can be used multiple times.
`--useAbsoluteTimeStamps`	Use absolute time stamps instead of relative time stamps.
`--userDataDir` *directory*	Specifies the user data directory.
(`-v` | `--version`)	Display version information and exit.
HLA Only	
(`-a` | `--appNumber`) *number*	Application number.
(`-f` | `--fomMapperLib`) *libname*	FOM Mapper library name.
(`-F` | `--fedFileName`) *fedfile_name*	FED file name.
`--fomMapperInitData` *data*	FOM Mapper initialization data.
`--fomModules` *module*	Specifies a FOM Module. Can be used multiple times. (HLA Evolved only.)
(`-H` | `--hostAddressString`) *address*	Specifies the host address.
(`-i` | `--sessionId`) *ID*	Specifies the session ID for this instance of the front-end.
(`-N` | `--federateName`) *federate_name*	Federate name. Default: VR-Forces. (HLA Evolved only.)
(`-p` | `--federateType`) *type*	The federate type distinguishes different categories of federates.
`--rprFomRevision`	RPR FOM revision.
`--rprFomVersion` *version_number*	RPR FOM version. (0.5, 0.7, 0.8, 1.0, 2.0006, 2.0014, or 2.0017)
(`-s` | `--siteId` *ID*	Site ID.
(`-S` | `--localSettingsDesignator`) *designator*	Specifies a string that is passed to the RTI during federate connect. (HLA Evolved only.)
`--ignoreAdvisories`	Enables or disables HLA advisory messages. Default: true.
`--useGeographicDdm`	Specifies use of geographic DDM if you are using the MÄK RTI. Default: false.
(`-x` | `--execName`) *exec_name*	Execution name. Default: VR-Link.
DIS Only	
(`-a` | `--appNumber`) *number*	Application number.
(`-A` | `--destAddrString`) *address*	Destination address.
`--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: 5.
(`-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.
`--mcastTtl`	Specifies the multicast time-to-live. Default: -1.
`--multicastAddresses` *address1* ... *addressn*	Specifies one or more multicast addresses.
(`-P` | `--disPort`) *port*	DIS port.

VR-Forces Quick Reference Card

vrfGui Command-Line Options

Option	Description
`--recvBufferSize` *size*	Receive buffer size.
`(-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.

Mouse Button Mappings

Navigation Action	Mouse Action	Context
Attach to entity.	**Shift** + left-click	Entity.
Attach to prop.	**Ctrl** + **Shift** + left-click	Prop.
Deselect all.	Left-click	Terrain or sky.
Drag terrain.	Left-click + drag	Terrain or prop.
Object selected, others unselected.	Left-click	Selected entity or prop.
Select entity or prop.	**Ctrl** + left-click	Unselected entity or prop.
Select object and unselect all others.	Left-click	Unselected entity or prop.
Teleport to location.	**Shift** + left-click	Terrain.
Teleport to prop.	**Shift** + left-click	Prop.
Teleport to entity.	**Ctrl** + **Shift** + left-click	Entity.
Unselect.	**Ctrl** + left-click	Selected entity or prop.
Display context-sensitive menu.	Right-click	Entity or prop.
3D Only		
Drag the view.	Middle mouse button drag	All.
Drag heading and pitch (mouse look).	Right Drag	All.
Orbit drag.	**Ctrl** + left-click + drag	Terrain or prop.
Zoom in (magnification).	**Ctrl** + mouse wheel forward	All.
Zoom out (magnification).	**Ctrl** + mouse wheel backward	All.
Reset magnification.	Middle mouse button	All.
Move observer forward.	Mouse wheel forward	All.
Move observer back.	Mouse wheel back	All.
Move observer level observer forward.	**Shift** + mouse wheel forward.	All.
Move observer level observer backward.	**Shift** + mouse wheel backward.	All.
Move flashlight beam.	Right mouse button drag	All.

Copyright © 2015 VT MÄK. All rights reserved. Document ID: VRF-4.3.1-QRC-150625

Mouse Button Mappings

Navigation Action	Mouse Action	Context
Expand flashlight beam.	**Alt** + mouse wheel forward.	All.
Contract flashlight beam.	**Alt** + mouse wheel backward.	All.
2D Only		
Rotate terrain.	Right-click + drag	All. Orient North disabled.
2D Zoom in.	Mouse wheel forward; Middle button + drag	All.
2D Zoom out.	Mouse wheel backward; Middle button + drag	All.

K = numeric keypad (K8 means press Keypad 8). To use the numeric keypad for navigation, NumLock must be on.

Observer Keyboard Mappings

Observer Movement	Observer	Level Observer	Laptop	2D Level Observer
Move forward.	**w**	unmapped	**w**	unmapped
Move back.	**s**	unmapped	**s**	unmapped
Move left.	**a**	unmapped	**a**	unmapped
Move right.	**d**	unmapped	**d**	unmapped
Move up.	**q**, **Ctrl** + **q**	unmapped	**q**, **Ctrl** + **q**	unmapped
Move down.	**e**, **Ctrl** + **e**	unmapped	**e**, **Ctrl** + **e**	unmapped
Move forward level observer.	unmapped	**w**	unmapped	**w**
Move back level observer.	unmapped	**s**	unmapped	**s**
Move left level observer.	unmapped	**a**	unmapped	**a**
Move right level observer.	unmapped	**d**	unmapped	**d**
Move up level observer.	unmapped	**q**, **Ctrl** + **q**	unmapped	unmapped
Move down level observer.	unmapped	**e**, **Ctrl** + **e**	unmapped	unmapped
Move attached context forward.	**Shift** + **w**	**Shift** + **w**	**Shift** + **w**	**Shift** + **w**
Move attached context back.	**Shift** + **s**	**Shift** + **s**	**Shift** + **s**	**Shift** + **s**
Move attached context left.	**Shift** + **a**	**Shift** + **a**	**Shift** + **a**	**Shift** + **a**
Move attached context right.	**Shift** + **d**	**Shift** + **d**	**Shift** + **d**	**Shift** + **d**
Move attached context up.	**Shift** + **q**	**Shift** + **q**	**Shift** + **q**	unmapped
Move attached context down.	**Shift** + **e**	**Shift** + **e**	**Shift** + **e**	unmapped
Level yaw left.	K **4**	K **4**	Left arrow	Left arrow
Level yaw right.	K **6**	K **6**	Right arrow	Right arrow
Level pitch up.	**Shift** + **j**	**Shift** + **j**	**Shift** + **j**	unmapped
Level pitch down.	**Shift** + **k**	**Shift** + **k**	**Shift** + **k**	unmapped
Level roll right.	**Shift** + **i**	**Shift** + **i**	**Shift** + **i**	unmapped
Level roll left.	**Shift** + **m**	**Shift** + **m**	**Shift** + **m**	unmapped
Yaw left.	**Shift** + **h**	**Shift** + **h**	**Shift** + **h**	**Shift** + **h**
Yaw right.	**Shift** + **l**	**Shift** + **l**	**Shift** + **l**	**Shift** + **l**

VR-Forces Quick Reference Card

Observer Keyboard Mappings

Observer Movement	Observer	Level Observer	Laptop	2D Level Observer
Pitch up.	K **8**	K **8**	↑	unmapped
Pitch down.	K **2**	K **2**	↑	unmapped
Roll right.	K **9**	K **9**	K **9**	unmapped
Roll left.	K **3**	K **3**	K **3**	unmapped
Orbit right.	**Ctrl**+**d**, →	**Ctrl**+**d**, →	**Ctrl**+**d**	unmapped
Orbit left.	**Ctrl**+**a**, ¬	**Ctrl**+**a**, ¬	**Ctrl**+**a**	unmapped
Orbit up.	**Ctrl**+**w**, ↑	**Ctrl**+**w**, ↑	**Ctrl**+**w**	unmapped
Orbit down.	**Ctrl**+**s**, ↓	**Ctrl**+**s**, ↓	**Ctrl**+**s**	unmapped
Look up.	**Ctrl**+K **8**	**Ctrl**+K **8**	**Ctrl**+K **8**	**Ctrl**+K **8**
Look down.	**Ctrl**+K **2**	**Ctrl**+K **2**	**Ctrl**+K **2**	**Ctrl**+K **2**
Look left.	**Ctrl**+K **4**	**Ctrl**+K **4**	**Ctrl**+K **4**	**Ctrl**+K **4**
Look right.	**Ctrl**+K **6**	**Ctrl**+K **6**	**Ctrl**+K **6**	**Ctrl**+K **6**
Look tilt left.	**Ctrl**+K **3**	**Ctrl**+K **3**	**Ctrl**+K **3**	**Ctrl**+K **3**
Look tilt right.	**Ctrl**+K **9**	**Ctrl**+K **9**	**Ctrl**+K **9**	**Ctrl**+K **9**
Zoom In	**=**	**=**	**=**	**e**,**=**
Zoom Out	**-**	**-**	**-**	**q**, **-**
Suppress constrain to terrain.	**x**	**x**	**x**	**x**
Reset observer:	**Space**	**Space**	**Space**	**Space**
In absolute mode, go to the center of the scene. If attached, reset the attach position and orientation.				
Level observer.	K **5**	K **5**	**Shift**+**Space**+ ↓	K **5**
Toggle Force North Orientation	unmapped	unmapped	unmapped	unmapped
Toggle Orient To Target Constraint	unmapped	unmapped	unmapped	unmapped
Reset the camera.	K **0**	K **0**		K **0**
Detach.	**z**	**z**	**z**	**z**
Attach to the next entity allowed by the attachment filter.	Period (**.**)	Period (**.**)	Period (**.**)	Period (**.**)
Attach to next entity or prop.	**>**	**>**	**>**	**>**
Attach to the previous entity allowed by the attachment filter.	Comma (**,**)	Comma (**,**)	Comma (**,**)	Comma (**,**)
Attach to previous entity or prop.	**<**	**<**	**<**	**<**
Next attach mode.	**Shift**+**z**	**Shift**+**z**	**Shift**+**z**	**Shift**+**z**
Increase linear speed.	K **7**	K **7**	K **7**	K **7**
Decrease linear speed.	K **1**	K **1**	K **1**	K **1**
Increase orbit speed.	K **+**	K **+**	K **+**	K **+**
Decrease orbit speed.	K **–**	K **–**	K **–**	K **–**
Toggle speed scaling on or off.	K *****	K *****	K *****	K *****
Toggle constrain to terrain on or off.	**/**	**/**	**/**	**/**

Copyright © 2015 VT MÄK. All rights reserved. Document ID: VRF-4.3.1-QRC-150625

Observer Keyboard Mappings

Observer Movement	Observer	Level Observer	Laptop	2D Level Observer
Clear the current movement mode. This key is an escape mechanism if keys are not behaving as you expect. This is typically due to a loss of the window context.	**c**	**c**	**c**	**c**
Toggle view controls on or off.	**Shift**+**c**	**Shift**+**c**		**Shift**+**c**
Toggle entity labels on or off.	unmapped	unmapped	unmapped	unmapped

vrfLauncher Command-Line Options

Option	Description
`(-B | --backend)`	Start vrfLauncher for the back-end only.
`B--`	Following `--`, arguments after `B--` are passed only to the back-end.
`(-C | --config)`	Open the Simulation Connections Configuration tool.
`-debug`	Following `B--` or `F--`, appends a d to vrfGui or vrfSim to run the debug version. (Windows only)
`(-F | --frontend)`	Start vrfLauncher for the front-end only.
`F--`	Following `--`, arguments after `F--` are passed only to the front-end.
`(-h | --help)`	Displays help information in a console window.
`--usePredefinedConnection` *connection*	Specify a connection name, such as DIS localhost, so that the vrfLauncher can launch without needing to display the Simulation Connections dialog box.
`(-v | --version)`	Displays version information.
`--`	Pass all arguments after this to vrfGui and vrfSim, except per `B--` or `F--`.



Link - Simulate - Visualize

VRF-4.3.1-1-150630

150 CAMBRIDGE PARK DRIVE, 3RD FLOOR CAMBRIDGE, MA 02140 617.876.8085 www.mak.com