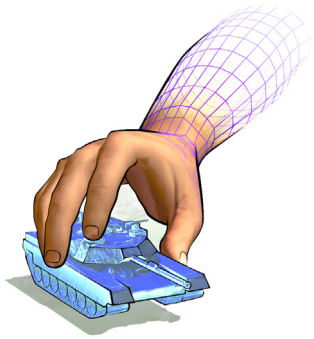

Entity Boost for VR-Forces

Entity Boost for VR-Forces is a plug-in to VR-Forces 3.12 that adds support for HLA Data Distribution Management (DDM).

Document ID: VRF-3.12-15-091117

Introduction

Entity Boost for VR-Forces is a plug-in module for VR-Forces that enhances its ability to simulate very large scenarios over multiple VR-Forces simulation engines. Entity Boost for VR-Forces uses HLA Data Distribution Management (DDM)[1]. Using DDM, you can add VR-Forces simulation engines to linearly scale up the number of entities in a simulation.

Entity Boost for VR-Forces also can improve performance of the 2D front-end for large scenarios.

DDM is only useful when you are using two or more simulation engines.

DDM is part of the HLA standard. Therefore, you cannot use Entity Boost for VR-Forces in DIS exercises.

Operating Systems Supported

Table 1 lists the operating systems supported by Entity Boost for VR-Forces. Please see *VR-Forces Release Notes* for additional information about operating system, compiler, and compatibility issues.

Table 1: Platforms supported

Operating System	Compiler
Red Hat Enterprise Linux Workstation 4, 5	Default compiler
Windows XP	Microsoft Visual C++ 7.1
	Microsoft Visual C++ 8.0 SP1
	Microsoft Visual C++ 9.0 SP1 32 and 64 bit libraries*
Windows Vista	Microsoft Visual C++ 8.0 SP1
	Microsoft Visual C++ 9.0 SP1 32 and 64 bit libraries*

* 64 bit libraries are available only for *vrfSim* (simulation engine).

1. DDM is an HLA service that provides interest management. It allows federates that are part of a scenario to register interest in only a particular subset of the simulation, and then only receive and process updates that are relevant to that subset.

RTI Support

Entity Boost for VR-Forces should work with RTIs that meet the HLA 1.3 specification or the SISO DLC HLA API[2] 1516 (SISO-STD-004.1-2004) version of the IEEE 1516 specification and are built with the same compilers as Entity Boost for VR-Forces. If you use the MÄK RTI, you must use version 3.4a or later.

Installing Entity Boost for VR-Forces

Entity Boost for VR-Forces is a plug-in to VR-Forces 3.12.

To install Entity Boost for VR-Forces on Windows:

1. Install VR-Forces, as described in *VR-Forces Users Guide*.
2. Run the Entity Boost for VR-Forces executable. It is a typical Windows wizard-style installer.
3. You must install the Entity Boost for VR-Forces plug-in into your VR-Forces root directory. By default, the installer offers to install the plug-in in the default VR-Forces installation directory. If you have installed VR-Forces in some other location, be sure to select the correct installation directory.
4. Since you are installing the plug-in into an existing directory, the installer asks if it is OK to install. Click Yes.

Installing the Entity Boost for VR-Forces Plug-in on Linux

➤ To install the Entity Boost for VR-Forces plug-in on Linux, untar the file in your VR-Forces root directory.

If you use the vrfLauncher application to start VR-Forces, you must add the DDM connection configurations.

➤ To add the DDM connection configurations, run *./bin/migrateSettings*.

Uninstalling the Entity Boost for VR-Forces Plug-in

You can uninstall Entity Boost for VR-Forces on Windows from the Control Panel or by running *./plugins/EntityBoost/unins000.exe*. There is no uninstaller for the Linux version. You can delete the *./plugins/EntityBoost* directory and *./plugins/EntityBoost.mtl*.

When you uninstall Entity Boost for VR-Forces, the connection configuration setting does not get removed. If you want to, you can delete the DDM connection configuration in the Simulation Connection Configuration dialog box, or you can revert to the default connection configurations by copying *./backup/connectionsSettings.xml* to *./config/connectionsSettings.xml*.

2. Simulation Interoperability Standards Organization Dynamic Link Compatible High Level Architecture Application Program Interface

Getting Started with Entity Boost for VR-Forces

This section helps you get started with interest management with a minimum of effort. It assumes the you are using the MÄK RTI and default settings. You will run an example scenario on two back-ends.

If you are the type of person who likes to install new software and start using it right away, this is the section for you. If you like to understand the concepts behind an application before you start to use it, skip this section and start reading Introduction to Data Distribution Management (DDM), on page 15 and the sections after it. Come back to this section when you are ready.

For details about interest management and DDM, and how to design scenarios and configure Entity Boost for VR-Forces, please see the later sections in this manual.

To use Entity Boost for VR-Forces, you must:

- Enable DDM in your RTI. (For details, please see Configuring Your RTI for DDM, on page 29.)
- Run VR-Forces with the correct HLA connection configuration.
- Design scenarios that take advantage of interest management. (For details, please see Designing Scenarios to take Advantage of DDM, on page 18.)
- Optionally, configure interest management to fine tune performance and to visualize how interest management is working.

Install and Configure the MÄK RTI and Start the rtiexec

To use Entity Boost for VR-Forces, you must enable DDM in your RTI.

To configure the MÄK RTI for DDM, complete this procedure on each computer that you are using:

1. Install MÄK RTI 3.4a.
2. Choose **Start** → **Programs** → **MÄK Technologies** → **MÄK RTI 3.4** → **Configuration** → **RTI Chooser**. The RTI Chooser opens (Figure 1).

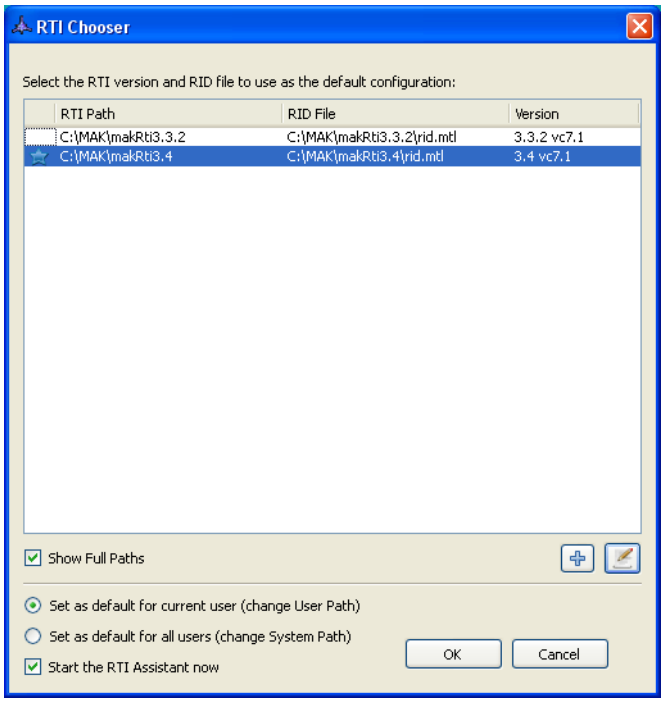

Figure 1. RTI Chooser

3. Click the Plus button. The Add RTI Configuration dialog box opens (Figure 2).

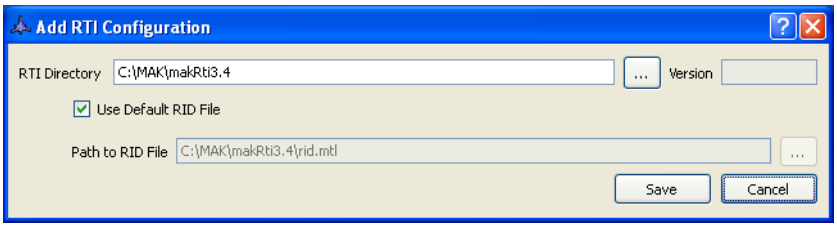

Figure 2. Add RTI Configuration dialog box

4. Clear the Use Default RID File check box.
5. Click the Browse (...) button next to the Path to RID File text box. The Select Desired RID File dialog box opens.
6. Select *ddm-rid.mtl*.

7. Click Open. The path to *ddm-rid.mtl* is added to the Add RTI Configuration dialog box .

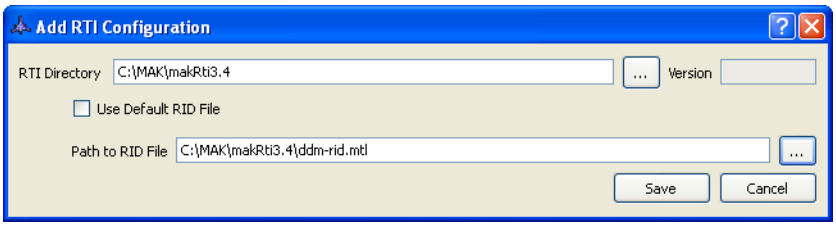

Figure 3. Add RTI Configuration dialog box

8. Click Save. The configuration is added to the RTI Chooser (Figure 5).

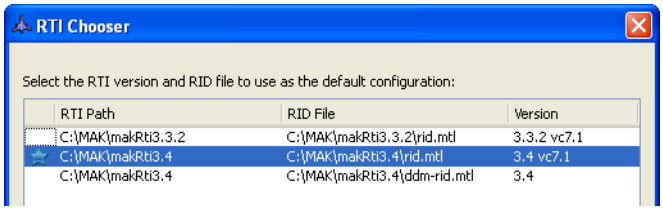

Figure 4. RTI Chooser with ddm-rid.mtl entry

9. In the RTI Chooser window, select the entry for *ddm-rid.mtl*.
10. Click OK. The MÄK RTI is now configured to use a RID file that has DDM enabled.

Start the rtiexec

Start the MÄK RTI rtiexec on one of the computers that you are using. It does not matter which computer you run it on.

This procedure assumes that there are no other MÄK RTI connections in use on your network. If this is not true, you may have to create a unique connection to use. Please see *MÄK RTI Users Guide* for details.

To start the rtiexec:

1. Choose **Start** → **Programs** → **MÄK Technologies** → **MÄK RTI 3.4** → **rtiexec**. The Choose Connection Configuration dialog box opens (Figure 1).

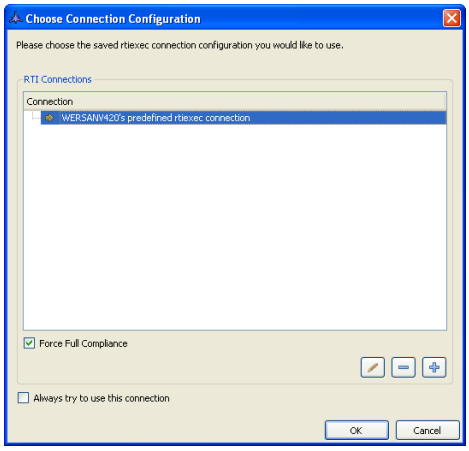

Figure 5. Choose Connection Configuration

2. Select the predefined connection for your computer.
3. Click OK. The rtiexec starts. (It runs as a daemon. You will not see anything.)

Run VR-Forces

This procedure explains how to start VR-Forces using HLA 1.3.

1. Start a front-end and back-end on one computer:

   On Windows, on the Start menu, choose **Programs** → **MAK Technologies** → **VR-Forces** *x.x* → **VR-Forces 2D GUI + Simulation Engine**.

   On Linux, in a console window, enter the following command:

   ```
   vrfLauncher --app vrfGui -z vrfSim
   ```

   The VR-Forces Simulation Connections Configuration dialog box opens.

   Select HLA 1.3 RPR 2.0 DIGuy DDM as the connection type and click Launch.

   You are prompted to select an RTI Connection for the front-end and the back-end (once for each). Select the connection for the rtiexec that you are running.

   The front-end and simulation engine 1:3001 start.

2. Start a back-end on the second computer:

   In a console window, change directory to *./binHLA1.3*. Enter the following command:

   ```
   vrfSim -F VRL-DIGuy20017DDM.fed -x VRL-DIGuy20017DDM
     --rprFomVersion 2.0017 -a 3002
   ```

   You are prompted to select an RTI Connection. Select the connection for the rtiexec that you are running.

   Simulation engine 1:3002 starts.

Running the Example Scenario in HLA 1516

To run the example scenario using HLA 1516 follow the instructions for using HLA 1.3, except for the following differences:

- In the VR-Forces Simulation Connections Configuration dialog box, select HLA 1516 RPR 2.0 DIGuy DDM.
- When you start the second back-end, change directory to *./binHLA1516*. Enter the following command:

  ```
  vrfSim -F VRL-DIGuy20017DDM.xml -x VRL-DIGuy20017DDM
    --rprFomVersion 2.0017 -a 3002
  ```

Run a Scenario Designed for Interest Management

Entity Boost for VR-Forces includes an example scenario. It is designed to work with two back-ends. The scenario contains approximately 1000 aggregate and individual entities. They are divided equally between the two back-ends. The entities simulated by each back-end are geographically isolated from one another. When DDM is enabled, entities simulated by one back-end are not known to the other back-end. (For details about why this is so, please see Introduction to Data Distribution Management (DDM), on page 15 and the other conceptual sections at the end of this manual.)

➤ To run the example scenario, in the front-end load *./data/scenarios/EntityBoostExample/EntityBoostExample_2_backends.scn*.

Figure 6 illustrates the scenario with entities from both simulation engines displayed. It shows which entities are simulated on back-end 1:3001 and which are simulated on back-end 1:3002.

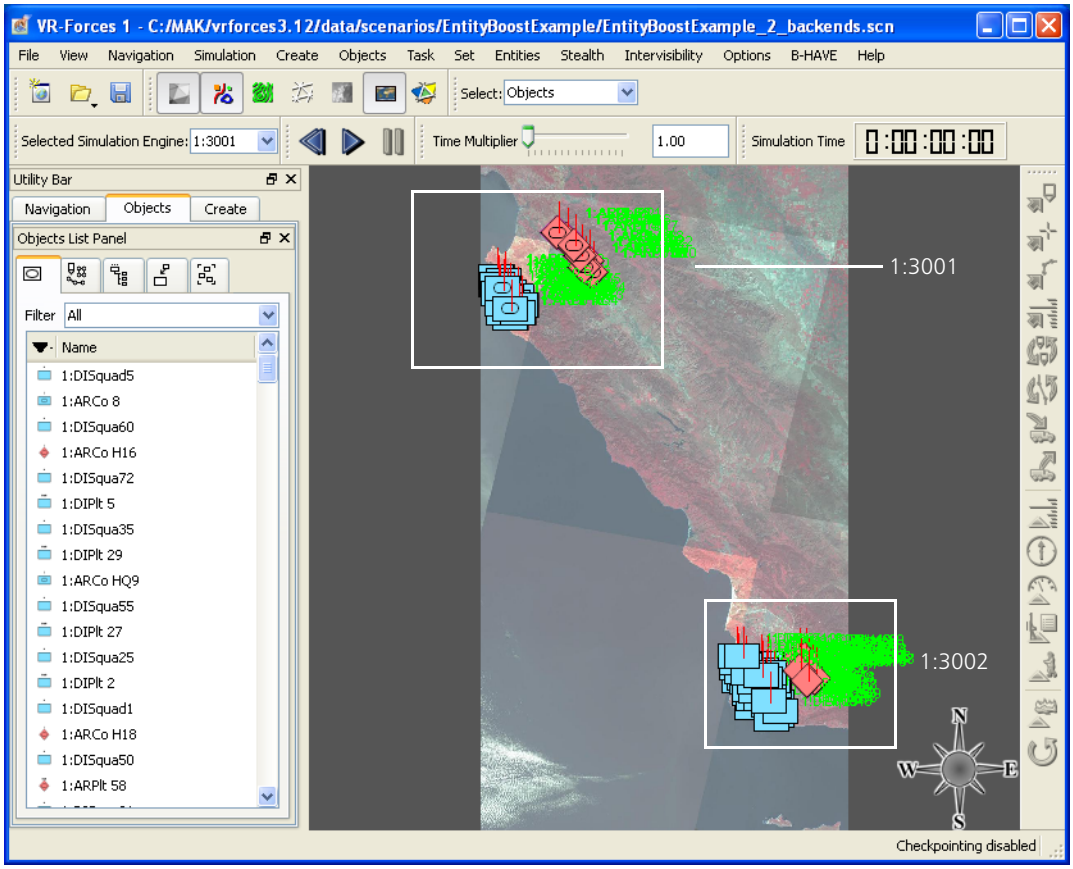

Figure 6. Example scenario

Visualizing Subscription Regions

As described in Visualizing DDM Subscription Regions, on page 22, the front-end can display the DDM regions that the back-ends are currently subscribed to.

To visualize subscription regions:

1. Choose **Options** → **Interest Management**. The Interest Management dialog box opens (Figure 7).

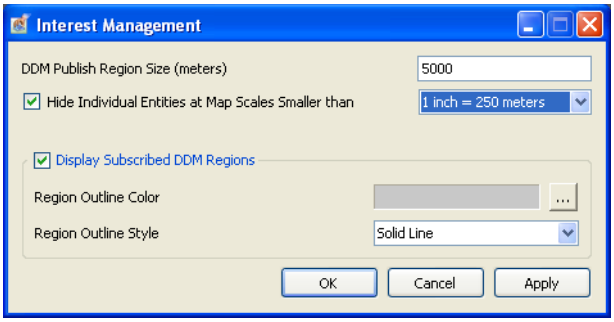

Figure 7. Interest Management dialog box

2. Select Display Subscribed DDM Regions
3. Click OK.

The map area displays rectangular boxes that show the subscribed regions for each back-end (Figure 8). In this scenario, there is no overlap between back-ends in the subscribed regions (only a single back-end is identified with each region). This is because the back-ends are geographically isolated from one another in this scenario. Note that the subscribed regions are not all continguous.

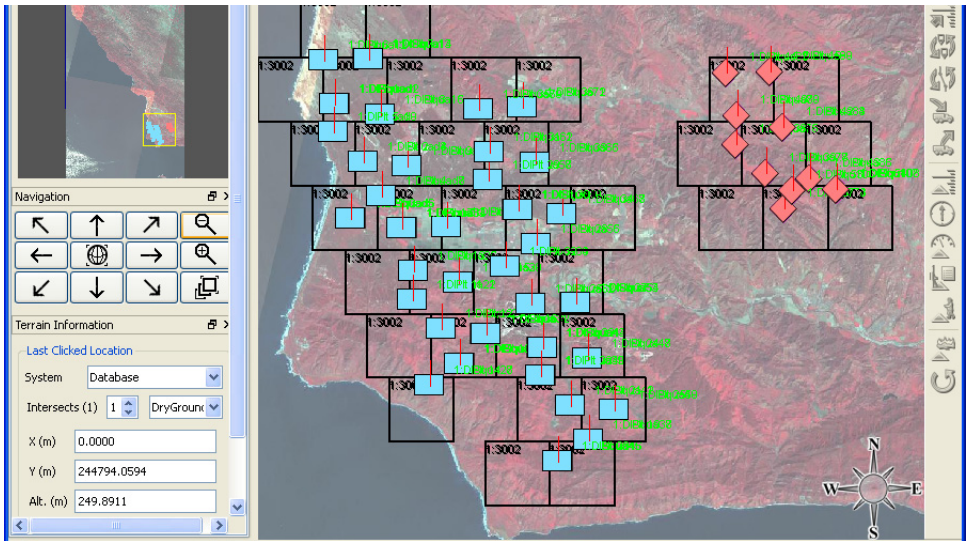

Figure 8. Subscription regions for back-end 1:3002

Controlling the Front-End's Entity Processing

Because there are large number of entities and aggregates in this scenario, you may want to limit how many objects the front-end has to be aware of and display at any given time. Reducing the numbers of objects (aggregates or individuals) that the front-end has to process and display can improve its responsiveness. This section describes ways to reduce the number of entities that the front-end has to process.

Filtering Out Individual Entities Based on the Map Scale

By default, the front-end displays all entities that are being simulated within the area of the map view. In part, this is to ensure that when you create new entities they are visible on the map. However, once you start running a simulation, this is probably not a concern, so you may want to take advantage of Entity Boost for VR-Forces's ability to hide individual entities when the view is zoomed out to a specific map scale.

To hide individual entities when zoomed out behond a specified map scale:

1. Choose **Options** → **Interest Management**. The Interest Management dialog box opens (Figure 7).
2. Select the Hide Individual Entities at Map Scales Smaller Than check box.
3. Click OK.

All individual entities disappear from the map area, the Objects List Panel and from the Object Count toolbar. Only aggregates are visible.

Specifying the Map Scale

The previous procedure described how to enable entity filtering based on the map scale. This procedure explains how to specify the map scale.

To control the map scale level at which individual entities are displayed:

1. Choose **Options** → **Interest Management**. The Interest Management dialog box opens (Figure 7).
2. Select Hide Individual Entities at Map Scales Smaller Than.
3. Select the map scale at which you want to see individuals.
4. Click Apply or OK.

Try zooming in and out of the map close to those areas containing entities and aggregates. Once you zoom in to a map scale value larger than the threshold you set, individuals start to appear in the map area, Objects List Panel, and Object Count widgets. As you zoom out to values smaller than that level, the individuals disappear.

Figure 9 illustrates this effect. The map scale is set to 1 inch = 5000 meters. In view 1, several aggregate entities are expanded. Four individual entities are below the aggregates. (The four individual entities are not part of the example scenario. They were added for this illustration.) In view 2, the map is zoomed out past the specified map scale. The aggregate entities collapse and the individual entities are not displayed. In view 3, the map is zoomed back in below the specified map scale. The individual entities are displayed. (The aggregates do not automatically expand, but could be expanded at this map scale.)

Figure 9. Hiding and showing individual entities by map scale

Filtering Out Entities by Changing the Map View

The previous section discussed how Entity Boost for VR-Forces filters entities in the front-end based on the map scale. Entity Boost for VR-Forces also filters entities based on the map view. Any entity that is not being simulated within the map view is automatically filtered out.

This feature creates some challenges for scenario creation. For details, please see Using DDM can Cause Problems when Assigning Task Destinations, on page 20.

1. Expand your view of the scenario to show the whole terrain (similar to Figure 6). Take a look at the list of objects. It is quite long.
2. Zoom in on the entities being simulated by back-end 1:3001. The entity list is shorter now. It only lists the entities in the current view. However, the list is still quite long .

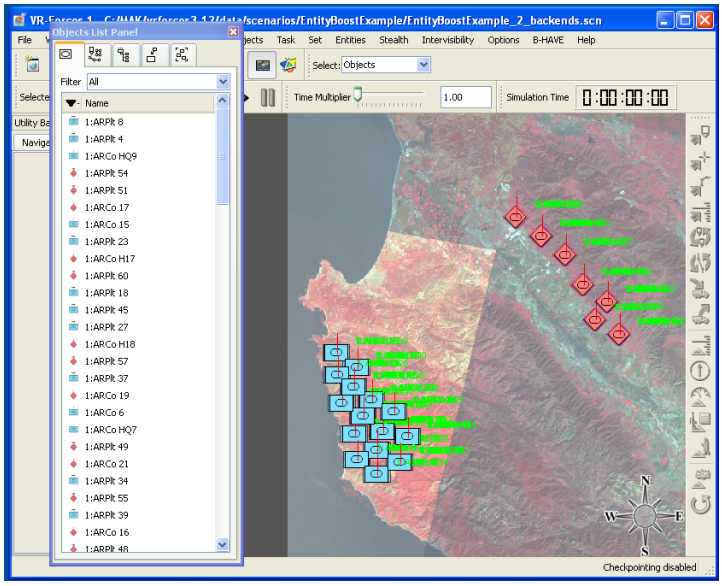

Figure 10. Entities on back-end 1:3001

3. Zoom in further to show just a few entities, as in Figure 11. Notice that the object list is now quite short. The front-end only knows about the entities being simulated within the extents of the map view.

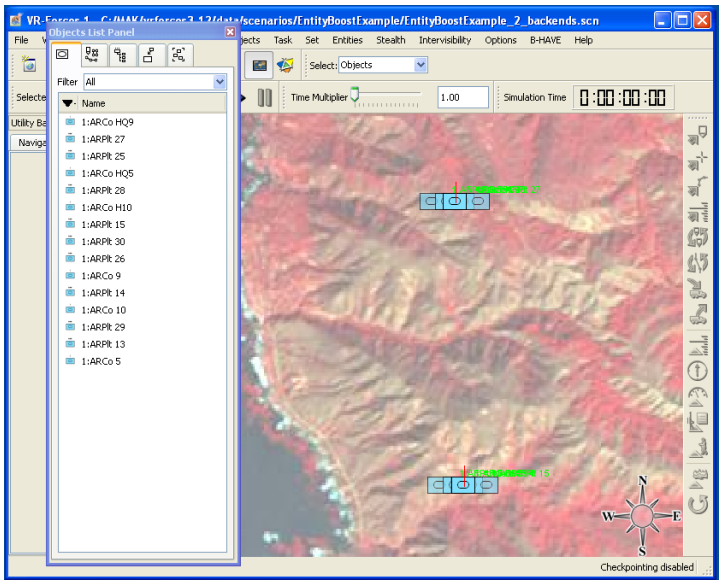

Figure 11. Object list restricted to entities in current view

Introduction to Data Distribution Management (DDM)

DDM is an HLA service that provides interest management. It allows federates that are part of a scenario to register interest in only a particular subset of the simulation, and then only receive and process updates that are relevant to that subset. In very large simulations, the overhead of receiving and processing updates from remote entities typically becomes the limiting factor in performance for all the federates participating in the simulation. Therefore, limiting updates to only a relevant subset allows simulations to scale dramatically.

The DDM capability in VR-Forces works by allowing VR-Forces to partition a scenario geographically among multiple VR-Forces simulation engines in such a way that there is minimal interaction between them. Each simulation engine does not need to have the entire state of the simulation in memory. It only needs the part of the simulation that it is concerned with. By dividing up the simulation in this way, each simulation engine is basically running its own smaller scenario, and yet they are all part of the larger simulation.

Geographic Partitioning

The partitioning of the scenario is done based on geography. Each simulation engine determines the portion of the earth that it is concerned with based on the position and capabilities of the entities that it is simulating. For example, if a simulation engine is simulating a human with a maximum sight distance of 1000 meters, then that simulation engine will subscribe to a region of the world that includes the 1000 meters around that entity. Remote entities that are not within that region of the world will not be received or processed by the simulation engine. Remote entities that are within 1000 meters are processed normally. As the locally simulated entity moves, the region of the world that the simulation engine subscribes to changes accordingly.

A simulation engine can subscribe to multiple, non-contiguous regions of the world. If a simulation engine is simulating two entities that are distant from each other, it subscribes to the regions around each of those entities, and not the regions between.

Figure 12 illustrates geographic partitioning. In part A, simulation engine 1 is publishing the rectangular (blue) entities. It subscribes to the area of the simulation that encompasses these entities (the solid-lined box). Simulation engine 2 is publishing the diamond-shaped (red) entities. It subscribes to the area of the simulation that encompasses these entities (the dashed-line box). Each simulation engine knows only about the three entities in its geographic area. The simulation engines do not know anything about the entities outside of their areas.

In part B of Figure 12, one of the entities owned by simulation engine 2 has moved over the terrain so that it is near the entities owned by simulation engine 1. Because this entity is no longer close to the other entities simulated by simulation engine 2, the simulation engine has subscribed to a second region of the terrain. One of the blue entities is within the subscription region of simulation engine 2, so it now knows about four entities. Similarly, because the red entity has entered simulation engine 1's subscription region, it also knows about four entities.

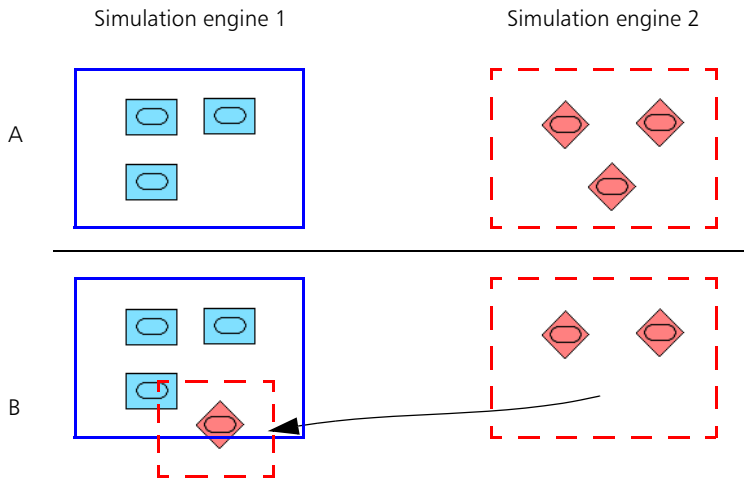

Figure 12. Geographic partitioning

DDM in the VR-Forces 2D Front-End

Entity Boost for VR-Forces also affects the 2D front-end. When scenarios get very large, the front-end can get overwhelmed by receiving and processing all the entity updates that come over the network. When DDM is enabled, the 2D front-end subscribes only to the area of the world that is within the 2D map display. Therefore, it only receives and processes the entities in that area. As you pan and zoom the map, the front-end changes it subscription region.

If you zoom out enough, the front-end stops subscribing to entity updates completely, and only receives aggregate updates. This means that at a certain map scale, the map will show only aggregate icons. You will need to zoom in to see the entity icons.

Object Name Prefixes for DDM-Enabled Scenarios

To ensure that all objects have unique names, VR-Forces uses object name prefixes when it names new entities in DDM-enabled scenarios.

In non-DDM scenarios, all back-ends are aware of the entities being created locally and on all other back-ends. When you create a new entity, the back-end on which it is created can give it a unique name because it knows the names of all other entities in the scenario. However, in a DDM-enabled scenario, a back-end does not necessarily know the names of all other entities in the scenario. (In fact, if the scenario is properly designed to take advantage of DDM, a back-end will certainly not know the names of all other entities.) Therefore, to ensure that there are no name conflicts between the entities created on different back-ends, VR-Forces assigns a unique number to each back-end and this number is prefixed to the name of each entity the back-end creates. For example, in a non-DDM scenario, entities simulated on two different back-ends might be named M1A2 1 and M1A2 2. In a DDM-enabled scenario, the entities might be named 1:M1A2 1 and 2:M1A2 1.

VR-Forces also ensures that the prefix values it assigns to a back-end will not conflict with the prefixes used in previously created scenarios.

- As always in VR-Forces, if you change an entity's name, you are responsible for ensuring that it does not conflict with another entity's name.
- Prefixes are assigned to the names of environmental objects only if you accept the default name. If you type a name in the object creation dialog box, it is not given a prefix.

Designing Scenarios to take Advantage of DDM

To take advantage of the benefits of DDM, when you create scenarios you must create entities that are near each other geographically, and which you expect to stay near each other, on the same simulation engine. The goal is to minimize the number of entities that each simulation engine must know about at any one time.

For DDM to provide the most benefit, each simulation engine must subscribe to the smallest geographic area possible, and there should be the least possible overlap between simulation engines.

For example, in the scenario illustrated in Figure 13, two battles are taking place – one in the western part of the terrain and one in the eastern part. Most entities will stay within the area of their particular battle. However, some air assets may be used in either part of the simulation.

The optimal DDM solution would be to place all the entities that will engage in the local battle in the west on simulation engine 1, place the entities in the local battle in the east on simulation engine 2, and place the aircraft, which may travel and participate in both battles, on simulation engine 3.

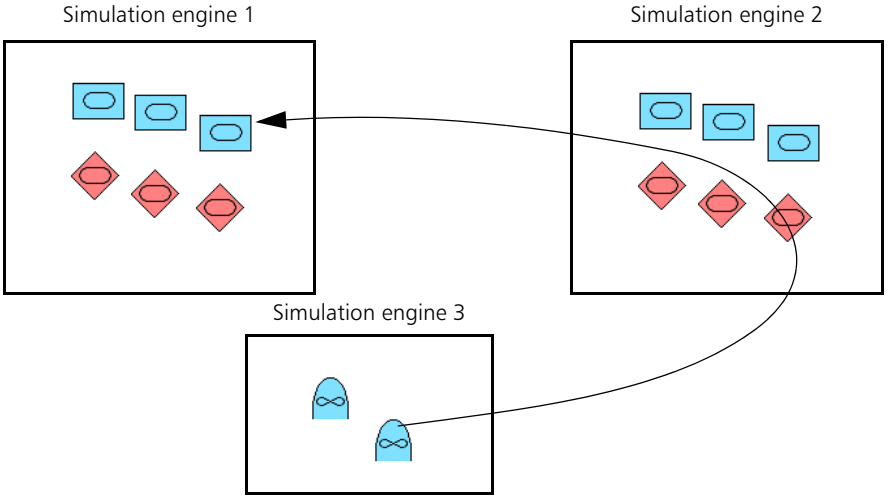

Figure 13. Optimizing entity assignment to simulation engines

Using the VR-Forces 2D Front-End with DDM

When you use DDM, the 2D front-end will usually not be subscribed to all the entities in the simulation at any one time. This means that entities listed in the entity lists, on the mini-map, and so forth will not be all of the entities in the entire simulation. As you pan the map and zoom in and out, entities will be added and removed from the entity lists and map view.

When the map is zoomed out beyond a particular map scale, the front-end can drop all information about individual entities, and only subscribe to aggregate entities. When this happens, the individuals are still being fully simulated in the VR-Forces simulation engines, but the front-end is simply not receiving the data. You can enable and disable this feature and you can configure the map scale at which individual entities are shown.

- Display of individual entities at all map scales is enabled by default so that when you create scenarios the individual entities that you create are displayed. (If display of individual entities is disabled and you are zoomed out beyond the map scale, you will not see newly created entities.) However, to take full advantage of DDM in the front-end, we recommend that you disable display of individual entities based on map scale when you run a scenario. For details, please see Configuring Interest Management at Runtime, on page 24.
- If individual entities are displayed when the map is zoomed too far out, the front-end may attempt to receive more entity updates than it can handle at a reasonable frame rate. If this happens, reduce the map scale at which individuals are shown.

Using DDM can Cause Problems when Assigning Task Destinations

Using DDM can cause the following problems for giving entities tasks that require specifying a control object or another entity:

- The front-end may allow you to specify invalid control objects or entities
- The front-end may not list valid control objects or entities.

Specifying Invalid Objects

A front-end can display entities and control objects that are being simulated by multiple back-ends. It includes all of these objects in its entity and object lists. Therefore, if you give an entity a task such as Move To Waypoint or Follow Entity, the list of objects to move to or entities to follow may include objects being simulated on back-end other than the one on which the entity you are tasking is being simulated.

However, when DDM is enabled, simulation engines are usually not aware of the objects being simulated on other back-ends. If you give an entity that is simulated on back-end A a task to move to a waypoint that is being simulated on back-end B and that waypoint or entity is not in back-end B's subscription region, the task will fail because back-end B does not know about the destination object.

For example, in Figure 14, M1A1 1 and Point 2 are both in the map view. Therefore, if you create a Move To Waypoint task for M1A1 1, Point 2 will be listed as a possible destination. However, since M1A1 1 is simulated by simulation engine 1 and Point 2 is simulated by simulation engine 2 and it is not in simulation engine 1's subscription range, it is not a valid destination and the task will fail.

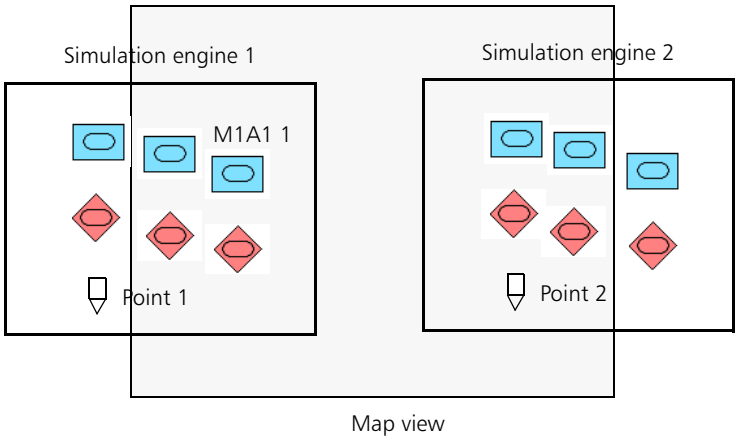

Figure 14. Invalid destination

Workarounds

There is no simple, guaranteed way to avoid the problem of invalid destination objects. However, you can minimize the likelihood of this problem as follows:

- Orient the map view to show objects that are all or mostly all from the same simulation engine. If you have designed your simulation with DDM in mind, then the objects on a given back-end are probably in the same geographic area and objects simulated on other back-ends are not in this area. If you set the map view to the area used by a particular back-end, then there should be few objects in the lists that are not valid for the entities you are giving tasks to.
- Before selecting an object as the target of a Move To, Follow, or similar task, check to see if it is being simulated on the same back-end and the entity being tasked. To find out which back-end an object is simulated on, display the Entity Information dialog box (for entities) or display the Entity ID in the Objects list (for entities and environmental objects). The entity ID includes the site and application ID for the back-end on which the object is simulated, for example 1:3002.

Specifying Unlisted Valid Objects

The front-end filters out information about all objects that are not within the map view. This means that it might not list objects that are valid destinations for a Move To or Follow task.

For example, in Figure 14, M1A1 1 and Point 1 are both being simulated by simulation engine 1. Point 1 is a valid destination for a Move To Waypoint task for M1A1 1. However, Point 1 is not within the map view. Therefore, it will not be listed as a destination in the Move To Waypoint dialog box.

Workarounds

You can deal with “missing” objects in the following ways:

- Orient the map view to show all of the objects on the relevant simulation engine. If you have designed your simulation with DDM in mind, then the objects on a given back-end are probably in the same geographic area. You can set the map to show these objects and continue to take advantage of DDM. If the objects on a back-end are widely dispersed, this strategy could cause the front-end to display more objects than it can handle.
- The destination object does not have to be selected in a list or on the map. If you know the name of the destination object, you can type it in the task dialog box.

Visualizing DDM Subscription Regions

The front-end can display the DDM regions that the back-ends are currently subscribed to. This feature shows how the underlying partitioning scheme works and can help you understand how to assign entities to back-ends and understand the effect of changing subscription ranges and region size.

Visualization of DDM regions increases message overhead and can affect the front-end's performance. Therefore, the feature is disabled by default. It is expected that the feature will be used during scenario creation and debugging, but not during the simulation.

DDM subscribed region data is displayed as polygonal areas drawn in the main map area (Figure 15). The regions are roughly square-shaped on projected terrain databases, and perfect squares on geodetic terrains. Each back-end publishes its own regions, and there may be overlap with other back-ends. You can configure the color and line style used to display the regions.

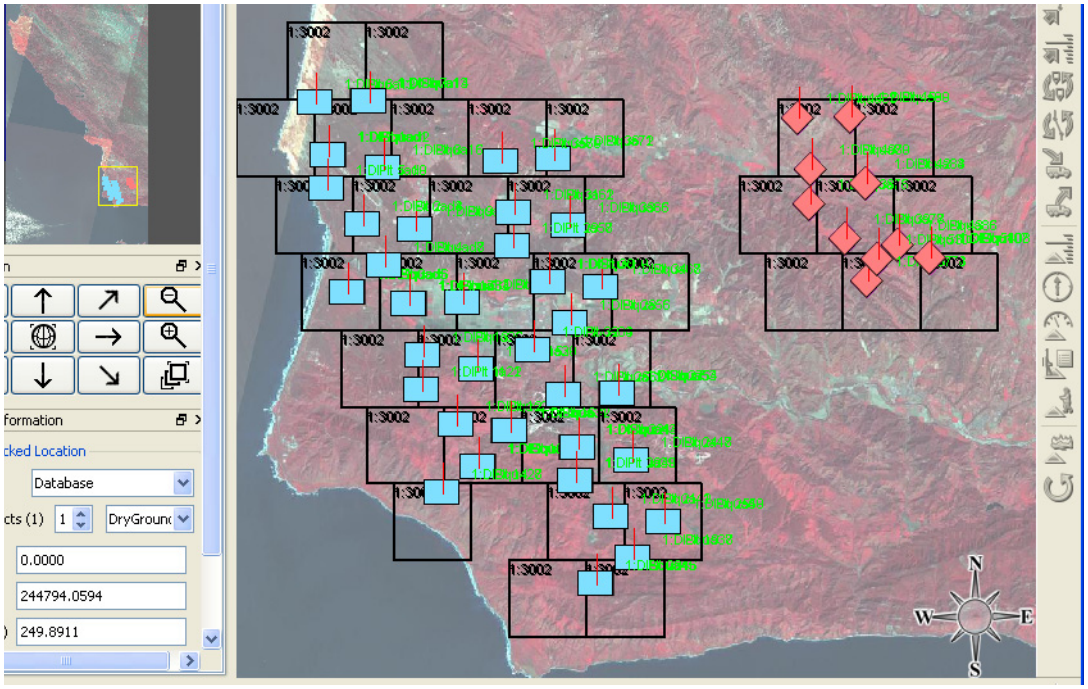

Figure 15. DDM subscription regions

Interacting with Other Federates

Since DDM is a scheme for making the communication between federates more efficient, the benefits of DDM are only seen when all federates in the simulation are using it. If a non-DDM federate is added to the simulation, it will receive all the updates for all the objects in the simulation continually, and may get overwhelmed by the data.

Since HLA is an open standard, and DDM is a service in HLA, DDM support can be added to any federate. The DDM scheme used by the federate must match the scheme used by VR-Forces. This scheme is described in The Entity Boost for VR-Forces DDM Scheme, on page 30.

Running VR-Forces with Entity Boost for VR-Forces

Entity Boost for VR-Forces requires a modified version of RPR FOM 2.0, draft 17 (specified by *VRL-DIGuy20017DDM.fed*). When you install Entity Boost for VR-Forces, it modifies the VR-Forces launcher to add configurations for HLA 1.3 and HLA 1516 that use the modified FOM. (Linux users must run a utility to do this. For details, please see Installing the Entity Boost for VR-Forces Plug-in on Linux, on page 4.) When you start VR-Forces, select HLA 1.3 RPR 2.0 DIGuy DDM or HLA 1516 RPR 2.0 DIGuy DDM. To start a back-end from the command line, a typical command would be:

```
vrfSim -F VRL-DIGuy20017DDM.fed -x VRL-DIGuy20017DDM
  --rprFomVersion 2.0017 -a 3002
```

Running VR-Forces without DDM when Entity Boost is Installed

When VR-Forces is configured to load the Entity Boost plug-in, it always tries to enable DDM at startup. However, VR-Forces can enable DDM only if your RTI is configured to support DDM and you specify the correct FOM. If VR-Forces determines that it cannot enable DDM, it displays a notification message. Then it starts with DDM disabled.

If you receive a notification message and you want to use DDM, make sure that you are using the correct FOM and that your RTI is configured for DDM, as described in Configuring Your RTI for DDM, on page 29.

If you are not using DDM and want to stop receiving the notification message, move *./vrforces3.12/plugins/EntityBoost.mtl* to a different directory or change its extension to something other than *.mtl*, so that the plug-in does not get loaded.

Configuring Interest Management in VR-Forces

The Entity Boost for VR-Forces plug-in has some configuration options that you can set at runtime and it has the following configuration files:

- *ddmConfigOptions.mtl*. Specifies performance options.
- *ddm_entity_type_subscription_range_list.mtl*. Specifies the subscription range for each entity type.

Entity Boost for VR-Forces can log DDM-related diagnostic information to file. It logs information such as the number, type, and duration of publish/subscribe region operations taking place each tick. This can be useful in troubleshooting performance problems. To enable DDM-related logging, set **logFrameRateStatistics** to 1 in *vrfSim.mtl*. A file named *ddmStats.<site>_<app>_<scenario name>.<unique integer>.log* is created for each separate scenario run.

Configuring Interest Management at Runtime

In the 2D front-end, you can configure the size of DDM regions and you can configure options for visualizing entities and DDM regions. Changes to region size do not affect the current scenario. Changes to visualization take effect immediately.

You can also configure subscription range. For details, please see Configuring a Back-End's Subscription Range, on page 27.

To configure interest management:

1. Choose **Options** → **Interest Management**. The Interest Management dialog box opens (Figure 16).

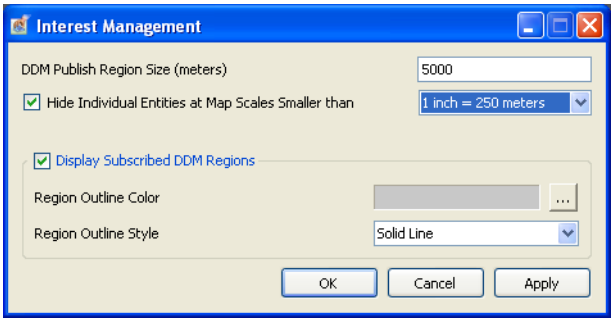

Figure 16. Interest Management dialog box

2. To change the DDM region size, enter a number, in meters, in the DDM Publish Region Size box. For guidelines, please see Setting DDM Region Size, on page 25 and Coordinating Region Size and Subscription Range, on page 28.

3. To disable display of entities when the map is zoomed out beyond a specified map scale, select the Hide Individual Entities at Map Scales Smaller Than check box.
4. To specify the map scale beyond which the front-end stops subscribing to individual entities, select a value from the Hide Individual Entities at Map Scales Smaller Than list.
5. To display DDM regions in the map view, select the Display Subscribed DDM Regions check box. (For more information about displaying DDM regions, please see Enabling the Display of DDM Regions, on page 25.)
    a. Optionally, specify the color of the region's outline by clicking the Region Outline Color browse button and selecting a color in the Select Color dialog box.
    b. Optionally, specify the line style of the region's outline by selecting an option on the Region Outline Style list.
6. Click OK.

Enabling the Display of DDM Regions

You can enable or disable the display of DDM regions from any front-end in a session. Changing the state in one front-end changes the state in all front-ends and back-ends in the session. The enable/disable state is maintained by the back-ends. When you change this status, the front-end sends a control message to the back-ends with the state change. All back-ends present on the network change their publication state and send out an updated data message containing the new status. Any other front-ends on the network receive this status and update their state.

Setting DDM Region Size

VR-Forces simulation engines use a fixed grid of geographic regions as the DDM regions. This is done to reduce the overhead and increase performance. By default, these regions are 5000 meters square.

To get the best performance in your large scenario, you may need to adjust the region size. You can change the region size in the Interest Management dialog box. Changes do not take effect until the next time a scenario is created or loaded.

Setting region size is a trade-off between granularity and performance, as follows:

- Smaller regions allow for a more fine-grained partitioning of the scenario. This is useful when your scenario has many entities in a small area, and multiple simulation engines are simulating different parts of the group.v
- There is processing overhead each time an entity crosses from one region to another. If your scenario has many fast moving entities, larger regions are better.

In most cases, the default region size will be sufficient, but you may be able to increase performance based on your specific scenario. For more information, please see Coordinating Region Size and Subscription Range, on page 28.

Editing Entity Boost for VR-Forces Configuration Files

The Entity Boost for VR-Forces plug-in has the following configuration files:

- *ddmConfigOptions.mtl*. Specifies performance options that affect subscription to, and unsubscription from, regions.
- *ddm_entity_type_subscription_range_list.mtl*. Specifies the subscription range for each entity type.

Table 2 describes the parameters in *ddmConfigOptions.mtl*. These parameters allow you to fine-tune subscription and unsubscription behaviors to suit your simulation.

Table 2: DDM configuration parameters

Parameter	Description
create-reflected-objects-threshold	The amount of time after which no more reflected object creations will be done in a single tick. Reflected object creations not completed in this tick occur in subsequent ticks. Default: 0.03 seconds.
max-publishing-regions	Specifies the maximum number of regions an object can publish to. If its extents are greater than this number, it publishes to all regions. Default: 10 regions. This parameter is meaningful only if publish-control-objects-to-world-region is set to 0. If you do not create or move control objects while a scenario is running, you probably do not need to change this parameter.
publish-control-objects-to-world-region	Specifies how to publish control objects. 0 — Publish to the regions they exist in. 1 — Publish to the playbox region (the world extent). Default: 1. A route or area may publish to many regions. If an object must publish to many regions, it can be quicker to publish to the entire world than to multiple specific regions.
region-timeout-threshold	Specifies how long VR-Forces waits, in wall-clock time, after the last object unsubscribes from a region, before it removes the region. Default: 30 seconds. Deleting unused regions prevents performance from degrading. However, if entities regularly move in and out of regions, it may be better to maintain them for a longer period of time rather than repeatedly creating and destroying them.

Table 2: DDM configuration parameters

Parameter	Description
subscription-adjustment-threshold	The amount of elapsed time, in seconds, beyond which VR-Forces will stop processing subscriptions and unsubscriptions. Unfinished subscriptions and unsubscriptions are queued for processing in subsequent ticks. Default: .03 seconds. In some cases the number of subscriptions and unsubscriptions per tick may result in unacceptably long tick times. This parameter lets you balance performance against the accuracy of subscriptions. (Delaying unsubscriptions should not affect the accuracy of the simulation.)

Configuring a Back-End's Subscription Range

For each entity that it simulates, a back-end subscribes to the region in which the entity is currently located, plus any adjacent regions within the subscription range of that entity. The subscription range configuration file, *./config/ddm_entity_type_subscription_range_list.mtl*, specifies the subscription ranges to use. The default file specifies subscription ranges for the following categories of entity:

- Ground platform — 5000 meters
- Air platform — 5000 meters
- Surface — 5000 meters
- Subsurface — 5000 meters
- Lifeform — 1000 meters
- Aggregate — 5000 meters.

You can change these generic subscription ranges and you can add entity-specific subscription ranges. A subscription range entry has the format:

```
(entity-subscription-range
  (entity-type  11 1 -1 -1 -1 -1 -1)
  (priority  5000.0)
)
```

To add a new subscription range, add a new block and change the entity type and value parameters. If you use wildcards (-1) in the entity enumeration, VR-Forces uses the best-match approach to map to specific entity types.

This configuration file can be different for each back-end.

Coordinating Region Size and Subscription Range

As you tune the performance of a scenario that uses Entity Boost for VR-Forces, bear in mind the relationship between region size and subscription range.

- If you decrease the region size without changing subscription ranges, you may increase the number of regions that a back-end has to subscribe to.
- If you increase an entity type's subscription range without changing region size, you may increase the number of regions that a back-end has to subscribe to.

Increasing the number of regions that a back-end has to subscribe to can affect performance, so try to keep region size and subscription range balanced. Remember that subscription ranges are set per entity type and can be different on each back-end. Region size is set for the entire scenario.

Configuring Your RTI for DDM

To use Entity Boost for VR-Forces, you must enable DDM. If you are not using the MÄK RTI, consult your RTI's documentation to determine how to enable DDM.

Configuring the MÄK RTI for DDM

If you are using the MÄK RTI, each back-end that runs in a DDM exercise must have a RID file that enables DDM. MÄK RTI 3.4a or later includes an example RID file, *ddm-rid.mtl*, that enables DDM. Be sure to configure each computer to use the customized RID file. For details, please see Section 4.1.1, "Search Order for RID Files", in *MÄK RTI Reference Manual*, or Chapter 8, *Configuring the MÄK RTI Startup Environment*, in *MÄK RTI Users Guide*.

MÄK RTI 3.4a has fixes to its DDM implementation that were made to support Entity Boost for VR-Forces. If you are using the MÄK RTI, you must use MÄK RTI 3.4a or later.

To enable DDM, set the following parameters to the values shown:

```
(setqb RTI_useRtiExec 1)
(setqb RTI_dataDistMgmt 1)
(setqb RTI_sendDiscoveredClass 1)
(setqb RTI_dualTransmitFirstInteractionRegions 1)
```

Optionally, you can enable hardware filtering of updates by specifying a mutually exclusive pool of multicast addresses for each back-end with the **RTI_minChannelAddr** and **RTI_maxChannelAddr** parameters, as follows:

```
;; assign first 65536 addresses to this BE (next BE should use
;; "224.1.0.0" and "224.1.255.255", then "224.2.0.0" and
;; "224.2.255.255" and so on)
(setqb RTI_minChannelAddr "224.0.0.1")
(setqb RTI_maxChannelAddr "224.0.255.255")
```

If you take this approach, you must customize the RID file for each computer that is running a back-end.

For more information about configuring the MÄK RTI for DDM, please see Configuring Distributed Region DDM and The Distributed Region Approach, in *MÄK RTI Reference Manual*.

The Entity Boost for VR-Forces DDM Scheme

In order for another HLA federate to interoperate with VR-Forces using DDM, that federate must implement a DDM scheme that is compatible with the scheme using by VR-Forces. This section describes the relevant parts of the VR-Forces DDM scheme that are necessary for another federate to comply with. This section assumes knowledge of HLA DDM. (MÄK *RTI Reference Manual* has as detailed discussion of DDM schemes.)

The VR-Forces DDM scheme uses geographic based regions. It used two dimensions, named "X" and "Y", which represent longitude and latitude respectively. The X or longitude dimension is a normalized value in the range from -180 degrees to 180 degrees. The Y or latitude dimension is a normalized value in the range from -90 degrees to 90 degrees.

Latitude and longitude values are normalized over a range of values when DDM regions are defined. The normalization range is different for HLA 1.3 and HLA 1516. Table 3 and Table 4 show the ranges for each dimension.

Table 3: X dimension (longitude) normalization

	–180 degrees	180 degrees
HLA 1.3	0	4294967195
HLA 1516	0	2147483547

Table 4: Y dimension (latitude) normalization

	–90 degrees	90 degrees
HLA 1.3	0	4294967195
HLA 1516	0	2147483547

Special Utility Regions

VR-Forces reserves the final 100 values in each dimension range to create utility regions. Only regions that comply with the special purpose of these utility regions, described below, should be created within these utility ranges.

The Out Of Bounds Publication Region

The Out of Bounds Publication Region is a special region in which objects can be published when they do not have a valid location. No federates should subscribe to this region, as it may contain many objects. In HLA 1.3, this region is a point region with the location:

X	4294967293
Y	4294967293

In HLA 1516, this is a region defined as:

X min	2147483645
X max	2147483646
Y min	2147483645
Y max	2147483646

The Out of Bounds Subscription Region

The Out of Bound Subscription Region is a special region to which federates can subscribe when they do not have a valid subscription area. No federates should publish any objects to this region. Federates that subscribe to this region should not receive any object updates. In HLA 1.3, this region is a point region with the location:

X	4294967291
Y	4294967291

In HLA 1516, this is a region defined as:

X min	2147483643
X max	2147483644
Y min	2147483643
Y max	2147483644

VR-Forces Object Publication

VR-Forces always publishes objects into a region that contains the object's location. Table 5 shows how the location of each object is defined.

Table 5: Object publication regions

Object Type	Publish Region
Entity	Region containing the entity's origin.
Aggregate	Region containing the aggregates's origin.
Control object (single point)	Region containing the point.
Control object (multi-point)	All regions within the extent of the object.
Fire interaction	Region containing the fire location.
Detonation interaction	Region containing the detonation location.
Entity specific data interaction	Region containing the origin of related entity.
Laser Designator	Region containing the designated location.

Control objects publish to at most 10 individual regions. By default, if their extent covers more than 10 regions, they publish to the playbox region. You can configure this value. For details, please see Editing Entity Boost for VR-Forces Configuration Files, on page 26.

Because changing region extents tends to be an expensive operation in HLA, VR-Forces uses static regions and moves objects to new DDM regions as they cross out of a defined region and into another.

VR-Forces Object Subscription

The VR-Forces 2D front-end subscribes to the geographical region that covers the viewing extent of the main map display.

A VR-Forces simulation engine subscribes to regions that are within the defined interest range of each locally simulated entity or object. The subscription ranges for different object types are defined as described in Configuring a Back-End's Subscription Range, on page 27.

Disaggregated entities and destroyed entities do not cause subscription.

Additional Ways to Improve Performance

The following actions can improve performance in addition to the benefits from using interest management:

- Disable the sending of back-end console messages to the GUI. Set **sendBackendLogToNetwork** to 0 in *vrfSim.mtl*.
- When you run scenarios that have many entities, you may need to increase the timeout values for **scenarioTimeoutInterval**, **backendTimeoutInterval**, and **saveTimeoutInterval** in *vrfGui.mtl*. The default value for these parameters is 60 seconds. You may need to experiment to find the best (shortest) timeout interval that provides enough time for the back-ends to load and save the scenario.
- Turn off aggregate route publishing. (For details, please see Section 11.3.1, "Displaying and Hiding Path Routes", in *VR-Forces Users Guide*.)
- Turn off display of DDM regions. (For details, please see Enabling the Display of DDM Regions, on page 25.)
- Disable **logFrameRateStatistics** in *vrfSim.mtl*.

Known Problems

Entity Boost for VR-Forces has the following known problems:

- On Windows, you cannot start a 64-bit simulation engine from the Start menu and have it automatically load Entity Boost for VR-Forces. You must use the `--loadPlugin` command line argument to load the Entity Boost for VR-Forces plug-in, for example:

  ```
  vrfSim_64 --loadPlugin entityBoostSim_64
     -F VRL-DIGuy20017DDM.fed -x VRL-DIGuy20017DDM
     --rprFomVersion 2.0017
  ```

- When you display regions in the front-end, after approximately 1000 regions are subscribed to, there may be errors in how the regions are drawn. This is just a display problem. There are no errors in the actual regions.
- VR-Forces does not save the settings for the map scale at which entities are not displayed and the state of the Display Subscribed DDM Regions check box. When you restart the front-end, these settings use the default values.
- Changing the MÄK RTI RID file parameter **RTI_maxUdpPacketSize** to a smaller value can cause subscription region updates to not be sent.