VII. Plans

This section describes how to write plans.

Introduction to Plans

VR-Forces has two kinds of plans: global plans and individual plans. Both kinds of plans can assign tasks and set data requests to a simulation object. Plans can also contain commands that create or delete simulation objects and tactical graphics. The commands in a plan can be conditional; that is, they will be executed only if a condition is satisfied. 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 individual plan is a set of statements that order a particular simulation object to complete a sequence of tasks, change its state, or both. The simulation object carries out the tasks in order unless it is interrupted by an independent task or global command.

A simulation object owns its individual plan. Changing its name, echelon relationship, or force does not change its plan.

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

If a simulation object does not have a plan, it does not mean that it has nothing to do. A simulation object may be assigned tasks by its superior in the force hierarchy. It can also be assigned tasks interactively from the GUI. If you want a simulation object to take planned actions independently of those assigned by its superior, you need to add statements to its plan.

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

```
  Move-To Waypoint:"red point"
  Wait-Duration Seconds-To-Wait:1 min 30 sec
  Move-To Waypoint:"white point"
Set Engagement-Rules="fire-at-will"
```

By default, plans are locked when a simulation is running, but you can choose to make edits by clicking the Unlock button. When you edit a plan during a simulation and then click Apply or OK to apply your changes, however, the simulation object starts executing the plan from the beginning, which may cause its activities to be out of synchronization with the other simulation objects in the simulation.

2. Plan Concepts

This chapter describes plans and plan concepts.

2.1. Introduction to Individual Plans and Global Plans

A plan contains one or more statements that VR-Forces executes without human intervention. You build these statements in dialog boxes. VR-Forces writes the statement to the plan, so you do not have to worry about syntax errors in your statements when you first create them. However, you must understand the meaning of individual commands and parameters and the implications of how you order them, to make sure that a plan does what you want it to do.

VR-Forces supports two types of plans: individual plans and global plans. They have these characteristics:

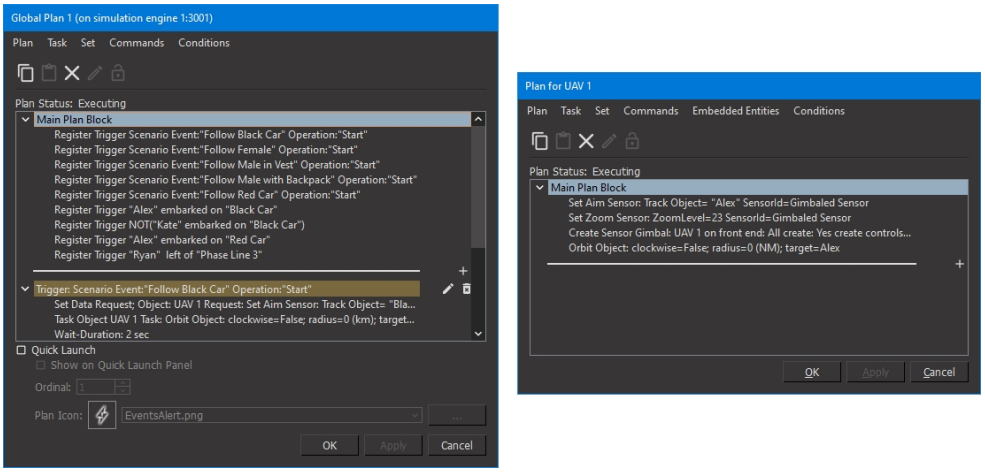
- Individual plans apply to a specific simulation object. They contain tasks and set data requests that are specific to the simulation object and are executed sequentially. The sequence of task execution can be interrupted by receipt of a task from an external actor (a global plan command or independent task). The task sequence can also be interrupted if a trigger (a form of conditional statement) or a reactive task is activated.

  > ⚠ Individual plans can include global commands (which are described later in this section). However, you must be careful not to use global commands, tasks, or sets for a simulation object in its own plan when you really want simulation object-specific tasks or sets. Sending a simulation object a global command, even from within its own plan, terminates its plan.

- Global plans are independent of any simulation object. The statements in a global plan affect the scenario as a whole (create and delete object commands) or a specific simulation object (task and set commands). Tasks sent to a simulation object from a global plan have the same effect on the simulation object as independent tasks assigned by a human player. They interrupt any task that the simulation object is engaged in and terminate its plan. Furthermore, if a global plan has a series of tasks for the same simulation object, they are sent in order without regard to completion of the previous task (as if you issued a series of commands from the Task menu, one immediately after another). In such a case, only the last command sent is likely to be completed.

Figure 1 shows a global plan and an individual plan. Note that the individual plan window (on the right) is tied to a particular simulation object (shown in the title bar).

Figure 1. Global plan and individual plan



2.2. Conditional Statements

Many of the statements used in plans set a parameter or direct a simulation object to perform a specific task unconditionally. However, the plan language also includes conditional statements. A conditional statement specifies a condition and a block of plan statements that are executed if the condition is true. VR-Forces supports these conditional statements:

- **If**. An **If** statement is evaluated once, when the plan execution flow arrives at the statement. If the condition is true at that point, VR-Forces executes the statements in the **If** block. For details, see "2.2.1 The If Statement" on page 7.
- Trigger. A trigger statement lets you plan for a condition without knowing in advance when the condition will occur. It specifies a condition that is evaluated continuously by VR-Forces. If the condition becomes true, VR-Forces executes the statements in the trigger. For details, see "2.2.2 Triggers" on page 7.
- **While**. A **While** statement executes a block of statements repeatedly while a condition remains true. VR-Forces evaluates the condition when the plan first arrives at the **While** statement. If the condition is true, it executes the statements in the block. When it completes the statements, it evaluates the condition again. If it is still true, VR-Forces executes the statements again. It continues to do so as long as the **While** condition is true. For details, see "2.2.3 While Statements" on page 9.
- **Do Until Interrupt**. A **Do Until Interrupt** statement allows you to assign a simulation object to perform a task until a condition takes place. Then the simulation object moves to the next statement in the plan. For details, see "2.2.4 Do Until Interrupt Statement" on page 10.
- **Wait Until**. A **Wait Until** statement causes a simulation object to wait in place until the condition becomes true. Then the simulation object moves to the next statement in the plan. For details, see "2.2.5 Wait Until Statements" on page 10.

The conditions that can be tested by conditional statements are:

- Is a simulation object in a given area? (Entity In Area)
- Is a simulation object within range of its target? (Entity in Range of Target)
- Is a simulation object to the left of a phase line? (Entity Left of Phase Line)
- Has a specific entity been created? (Entity Created)
- Is a simulation object destroyed? (Entity Destroyed)
- Is a simulation object under fire? (Entity Under Fire)
- Has a simulation object located a target? (Entity Has Target)
- Has a simulation object's embarkation state changed? (Entity Embarked)
- Is a simulation object above or below a specified altitude? (Entity Altitude)
- Has this simulation object detected another simulation object? (Detect Entity)
- Has this simulation object received a text message? (Receive Text Message)
- Has an engineering object been breached? (Engineering Object Breached)
- Has this simulation object received a missile approach warning? (Missile Approach Warning)
- Is a scenario event in process or concluded? (Scenario Event)
- Has a lifeform surrendered? (Lifeform Surrendered)
- Does a lifeform have a particular DI-Guy animation? (Entity DI-Guy Animation Check)
- Does a lifeform have a particular DI-Guy appearance? (Entity DI-Guy Appearance Check)
- Is a random probability true? (Random)
- Is a resource for this simulation object at a certain level? (Resource)
- Is a subordinate of a unit independently tasks? (Simulation Object Independently Tasked)
- Is the simulation time greater than or less than a specific elapsed time? (Simulation Time)
- Is the simulation time a certain value? (Simulation Time of Day)
- Is the simulation date and time a certain value? (Simulation Date/Time)
- Is a tactical graphic active? (Tactical Graphic Active)

Conditional statements can test for True and False, and can include the logical operators AND, OR, and NOT.

For details, see "2.2.6 Conditional Tests" on page 10.

2.2.1 The If Statement

An **If** statement tests a condition. It only tests the condition once.

> Do not use an **If** statement to test for receipt of a message or some other circumstance that does not represent a simulation object's state. VR-Forces does not keep a record of events and such a test will almost always evaluate to false. Use a trigger to test for conditions that do not evaluate simulation object state.

An **If** block consists of:

- The word **If**
- A condition to be evaluated
- A block of statements
- An **else** statement
- Optionally, statements for the **else** condition
- An **endif** statement

If the condition is true at the moment the **If** statement is evaluated, control passes to the first statement of the **If** block. If the condition is false, control passes to the first statement in the **else** block, if you have one, or to the first statement after the complete **If** block if you do not have an **else** block. You can nest **If** statements within **If** statements. The syntax is:

```
If (condition) then
  zero_one_or_more_statements
else
  zero_one_or_more_statements
endif
```

When VR-Forces reaches the end of the block of **If** or **else** statements, control passes to the first statement after the **If** statement.

The Add Conditional Expression dialog box ensures that you have the correct syntax. It does not ensure that your **If** block makes logical sense.

2.2.2 Triggers

A trigger consists of a condition and a block of statements that are executed when the condition becomes true.

When you create a trigger, it is not automatically added to the plan's execution sequence. You must register a trigger before VR-Forces will test it. (If you have a programming background, it may be helpful to think of triggers as functions or subroutines that you write separately and then call from elsewhere in a plan. Extending the analogy, a trigger is always local to its plan.)

When you register a trigger, VR-Forces checks the condition. If the condition is true, control passes to the first statement in the trigger body. If the condition is false, VR-Forces

continues to the next statement in the plan. It continues to check the trigger condition periodically. If at any point, the condition evaluates as true, control passes to the block of statements associated with the trigger.

When you create a trigger, you can specify that it interrupt the current task or that it suspend and resume the current task.

If you specify that a trigger interrupt the current task, task execution proceeds as follows:

- If a trigger contains any tasks that are mutually exclusive with the currently executing task, the current task is discarded and the statements in the trigger are executed.
- If the tasks in a trigger are not mutually exclusive with the current task, or if the trigger only has set data request statements, the trigger's statements are executed and the current task continues. For more information about this case, see "Using Triggers That Do Not Have Mutually Exclusive Tasks" on page 16.
- After the tasks in the trigger block are completed, control returns to the next statement in the simulation object's plan after the one that was interrupted by the trigger.

If you specify that a trigger suspend the current task, task execution proceeds as follows:

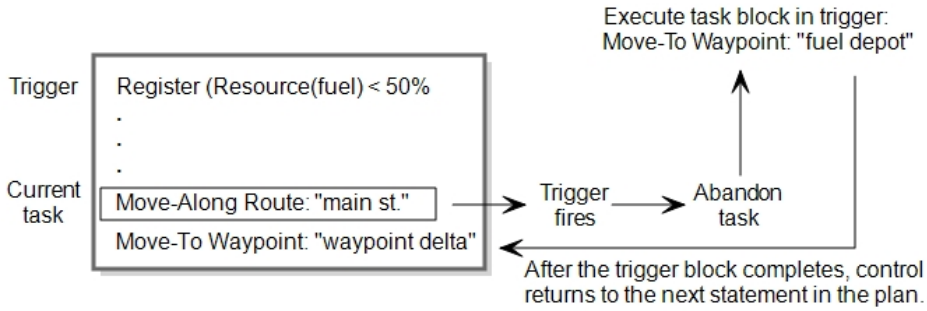
- If a trigger contains any tasks that are mutually exclusive with the currently executing task, the current task is suspended and the statements in the trigger are executed.
- If the tasks in a trigger are not mutually exclusive with the current task, or if the trigger only has set data request statements, the trigger's statements are executed and the current task continues. For more information about this case, see "Using Triggers That Do Not Have Mutually Exclusive Tasks" on page 16.
- After the tasks in the trigger block are completed, the suspended task is resumed, if possible. There may be cases where the previous task is no longer meaningful due to changes in the scenario.

> ⓘ The location of a trigger in a plan has an important effect on its functioning. For a discussion about where to place triggers, see "Deciding Where to Put Triggers" on page 16.)

By default, the name of a trigger is the syntax of its condition. However, you can change the name of a trigger to be any text you want to improve readability or shorten it.

Figure 2 illustrates the process when a trigger has a task that is mutually exclusive with the current task and you have specified that the current task be discarded. After the tasks in the trigger are completed, control returns to the next statement in the simulation object's plan. If the simulation object was in a **While** loop when the trigger fired, it returns to the next statement in the **While** block.

Figure 2. Flow of control for triggers



By default, after a trigger fires, it is removed from the list of registered triggers. If you want to continue to test for the same condition, you must reregister the trigger. You can design the plan in such a way that the trigger is registered again. Alternatively, when you create a trigger, you can specify that it reregister automatically.

> An alternative to using triggers in a plan is to write a reactive task. For information about reactive tasks, see "Creating Scripted Tasks and Sets".

2.2.3 While Statements

A **While** statement consists of:

- The word **While**
- A condition to be evaluated
- A block of statements
- An **endWhile** statement

If the condition is true at the moment the **While** statement is reached, control passes to the first statement of the **While** block. If the condition is false, control passes to the first statement after the complete **While** block. You can nest **While** statements within other conditional blocks.

When VR-Forces reaches the end of the **While** block, it evaluates the condition again. If the condition is still true, VR-Forces executes the statements again. If it is false, control passes to the first statement after the **While** block.

- VR-Forces re-evaluates a **While** condition only after it has executed all of the statements in the block. If the condition becomes false while VR-Forces is executing statements in the block, VR-Forces continues to execute all statements because it does not re-evaluate the condition until it completes the **While** block.
- If you put a continuous task such as Wait or Patrol Between in a **While** block, it is possible that the condition will never be re-evaluated, because these tasks never finish.

2.2.4 Do Until Interrupt Statement

A **Do Until Interrupt** statement allows you to have a simulation object perform a task, even one that otherwise could continue indefinitely. It consists of:

- The words **Do Until Interrupt**
- A condition to be evaluated
- An **End Do Until Interrupt** statement

If the condition is true at the moment the **Do Until Interrupt** statement is reached, the simulation object continues to the next task in the plan. If the condition is false, the simulation object performs the task(s) in the **Do Until Interrupt** statement. VR-Forces continually checks the condition until the condition becomes true or the simulation object is tasked by some other form of input, such as a trigger or independent task. If the condition becomes true, control passes to the next statement in the plan.

If a **Do Until Interrupt** state is interrupted by a trigger, the plan returns to the **Do Until Interrupt** task(s) after the trigger's tasks are completed.

2.2.5 Wait Until Statements

A **Wait Until** statement consists of:

- The words **Wait Until**
- A condition to be evaluated

If the condition is true at the moment the **Wait Until** statement is reached, the simulation object continues to the next task in the plan. If the condition is false, the simulation object waits in place. VR-Forces continues to check the condition periodically until the condition becomes true or the simulation object is tasked by some other form of input, such as a trigger or independent task.

If a **Wait Until** state is interrupted by a trigger and the trigger's Interrupt Current Task check box is not selected, the plan returns to the **Wait Until** state after the trigger's tasks are completed. If the trigger's Interrupt Current Task check box is selected, the plan moves to the statement after the **Wait Until**.

If the condition becomes true, control passes to the next statement in the plan.

2.2.6 Conditional Tests

This section describes each of the conditions that VR-Forces can test. When you set up conditions, keep in mind that:

- If there is no simulation object with the specified name, the result is always false.
- If the specified simulation object is destroyed, the condition might never be satisfied. Your plan should take this possibility into account.

Detect Entity

The Detect Entity condition tests to see if this simulation object has detected another simulation object at a particular identification level. The identification levels are Detected, Classified, Identified, and Full Knowledge. Identification levels are determined by the

detection probability configuration files, based on the distance between the detecting simulation object and the target and for how long the simulation object has been detected. For details about the detection probability files, see "Detection Tables".

> This condition is not supported in global plans.

Engineering Object Breached

The Engineering Object Breached condition returns true if the specified engineering object has been breached. (Aggregate-level scenarios only.)

Entity Altitude

Entity Altitude returns true if the simulation object (or member of a unit) is at a higher or lower altitude than the one specified.

Entity Created

Entity Created returns true when the specified simulation object is created in the scenario.

Entity Destroyed

Entity Destroyed returns true if the specified simulation object is destroyed.

- The definition of destruction for a unit is implementation specific. For the entity-level models provided with VR-Forces, units are considered to be destroyed when all members of the unit are destroyed.
- If you have implemented units locally using the VR-Forces Toolkit, they could have a different definition of destroyed.

Entity DI-Guy Animation Check

The Entity Di-Guy Animation Check condition tests the current DI-Guy animation for a lifeform. This condition may be useful if you are scripting a series of animations. For example, if a particular entity is using a threatening or hostile animation, you might want to have other simulation objects respond a certain way.

Entity DI-Guy Appearance Check

The Entity Di-Guy Appearance Check condition tests the current DI-Guy appearance for a lifeform. This condition may be useful if you are scripting a lifeform's interaction with another lifeform based on ethnicity or how they are dressed.

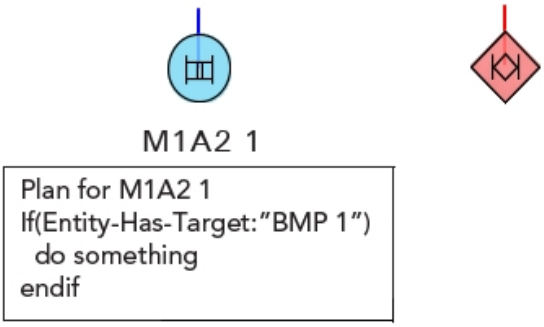
Entity Embarked

Entity Embarked returns true if the named simulation object is embarked.

Entity Has Target

Entity Has Target returns true if the simulation object specified in the condition identifies a target. For example, consider the entities in Figure 3. The condition in M1A2 1's plan is true when BMP 1 has a target, not when M1A2 1 targets BMP 1.

Figure 3. Entity Has Target condition



If a simulation object's rules of engagement prevent it from firing, this condition does not return true when the simulation object identifies a target.

Entity In Area

Entity In Area returns true if the simulation object reports that it is in the named area. You can use the **NOT** operator, to test whether a simulation object is outside the area.

If a simulation object enters an extruded area, VR-Forces does a 2D check first, to see if it is in the area. Then it checks the simulation object's altitude to see if it is within the area's volume.

If you have a plan in which you are testing for entities in an area and destroying them as they enter it, the Exclude Destroyed Simulation Objects option lets you reregister a trigger and only test for entities that have not been destroyed.

Entity In Range of Target

Entity In Range of Target returns true if the simulation object comes within the specified range of the selected simulation object.

Entity Left of Phase Line

Entity Left of Line returns true if the simulation object reports that it is to the left of the specified phase line.

A simulation object is considered to be to the left of a phase line if it is on the left side of an infinite extension of the phase line as viewed from a point on the line facing in the direction of the line. For a description of phase line direction, see "Phase Lines".

Entity Under Fire

Entity Under Fire returns true if the simulation object is under fire. An entity is considered to be under fire if it is directly targeted in a Fire PDU or if there is a hostile detonation within the **under-fire-parameter** distance specified in the **under-fire-determination-sensor** sensor in its OPE file. The default for all entities is 1000 meters.

Lifeform Surrendered

The Lifeform Surrendered condition tests to see if a lifeform entity has surrendered (using the Surrendered set data request).

Missile Approach Warning

The Missile Approach Warning condition returns true if a hostile missile is within the specified distance of this simulation object. This condition is not available in global plans.

Random

You can use the Random condition to randomly decide a simulation object's behavior at run time.

`Percentage` – A floating point probability percentage value between 0% (zero) and 100%. Random returns true if a random number between zero and 100 is less than or equal to the specified value.

If you use the Random condition, consider:

- Random is not very useful in triggers, since triggers may evaluate their condition expressions as often as once per tick. This means that a trigger expression of "Random Probability:50%" is likely to fire within the first couple of times it is evaluated. Therefore, we recommend that you not create triggers that are based on a random condition.
- If you use a random condition in a cascading sequence of **If** statements, and you want to ensure the correct probability distribution, be careful how you specify the percentage value. For example, randomly selecting one of three paths with equal probability would be done like this:

```
If (Random Probability:33.33%) then
  Move-Along Route:"1"
Else
  If (Random Probability:50%) then
    Move-Along Route:"2"
  Else
    Move-Along Route:"3"
  Endif
Endif
```

Receive Text Message

The Receive Text Message condition tests to see if this simulation object has received a text message that contains the text specified in the Receive Text Message dialog box. This condition is not available in global plans. The text message can be received as the result of a Send Text Message task or the Send Text Message command in a plan.

Resource

The Resource condition tests the amount of a specified resource for this simulation object using comparison operators. This condition is not available in global plans.

Scenario Event

The Scenario Event condition returns true if the specified scenario event is in process or has completed.

Simulation Object Independently Tasked

The Simulation Object Independently Tasked condition tests to see if an object that is a member of a unit is executing a task independently of the unit. If a unit member is independently tasked, you might not want to issue a new task to the unit until the subordinate finishes its task. This condition lets you test the independent task status of all members of a unit or particular members of a unit to help plan the timing of unit tasks. For example, in the Parachuting example scenario, the plan for the helicopter CH 47 1 waits until all members of Inf SQD 1 are not independently tasked before it sends a message to the squad.

Simulation Time

The Sim Time condition returns true if the elapsed simulation time is greater than (Sim Time >) or less than (Sim Time <) the time specified in the statement. Sim Time is the elapsed simulation time in days, hours, minutes, and seconds.

Simulation Date/Time

The Simulation Date/Time condition returns true if the simulation date and time matches the specified value.

Tactical Graphic Active

The Tactical Graphic Active condition returns true if a published tactical graphic is in the active state. For details about setting a tactical graphic's state, see "Active and Inactive Tactical Graphics" and "Activation".

Boolean Operators

You can use these boolean operators in conditional expressions:

- **True**.
- **False**.
- **AND** (*expression, expression*) - evaluates as true if both conditional expressions are true.
- **OR** (*expression, expression*) - evaluates as true if either of the expressions are true.
- **NOT** (*expression*) - evaluates as true if the conditional expression is false.

2.3. Plan Variables

Many tasks require you to specify a simulation object to act on, such as Fire on Target. Conditional tests, such as Entity Destroyed, require you to identify a simulation object to evaluate. However, you will not always know the name of the simulation object that you need to specify. Variables let you specify a simulation object without knowing its name. Plans can reference two variables:

- Simulation Object. The Simulation Object variable references a simulation object that has been identified by a conditional test, such as Entity in Area or Entity Left of Line. For example, suppose that if an opposing force entity enters an area you want

to fire at it. You can create a trigger using an Entity in Area condition. Then in the body of the trigger, you can insert a Fire at Target task that specifies the Simulation Object variable rather than the name of a specific entity.

In an AND condition, the variable is set to the last condition satisfied. In an OR condition, the variable is set to the first condition satisfied.

- Created Object. The Created Object variable references the most recently created object. For example, suppose you create an object from a plan and do not give it a name. You can issue it a plan or refer to it using the Create Object variable. This variable refers to the most recently created object.

Plan variables are specific to each simulation object plan. A variable value in one plan does not affect the variable value in a different plan. Therefore, if you want to use the Created Object variable, you must create the object in the plan in which you want to use it. You could not, for example, create an object in a global plan and then reference it using the Created Object variable from a simulation object's plan.

Once a variable has a value, you can use that variable in a plan as many times as you want to. However, you must keep track of plan logic that might change the value as the simulation object progresses through the plan.

For details about how to use variables in plans, see "3.5. Using Plan Variables" on page 37.

2.4. Viewing Plans

You can view a simulation object's plan and observe its progress through the plan. Statements are numbered to show the order in which they will execute unless interrupted by a trigger, independent task, or other external input. If a plan is executing the statements in a conditional block, VR-Forces draws a box around the block and the numbering is shown for the statements in the block.

If the Plan window is collapsed (that is, the Trigger pane is not visible) and control passes to a trigger, the window automatically expands to show the tasks in the trigger that is being executed.

The task that a simulation object is executing is highlighted in the Plan window. When the simulation object completes its plan, a message is displayed in the status area at the bottom of the Plan window.

You can view the plans for as many simulation objects at a time as you want. In addition to viewing plan status in the Plan window, you can view the current task on the Task Status page of the Information dialog box. The Last Selected Object Panel also lists the currently executing task.

➤ **To view a simulation object's plan:**

1. Select the simulation object whose plan you want to view.
2. Open the Plan window using one of the following methods:

- Choose **Objects > Plan**.
- Press **P**.
- On the Last Selected Object Panel or Command Panel, click the Plan button ().

2.5. Considerations for Creating Plans

This section has tips for constructing plans including how to avoid problems that are hard to debug and how statements affect each other, with implications for how to organize the statements in a plan.

2.5.1 Considerations for Using Triggers

Triggers are a versatile part of plans and you should have a good understanding of their intricacies before you use them.

Deciding Where to Put Triggers

VR-Forces does not evaluate a trigger condition until it registers the trigger. If you want to ensure that a trigger is always evaluated by VR-Forces, register it at the beginning of a plan before any non-trigger statements. If you register a trigger later in a plan, it is possible that it will never be evaluated. This could occur if a plan is abandoned. Of course, you might decide that you do not want a trigger to be evaluated unless a set of preceding tasks have been completed. In that case, placing it later in the plan may be exactly what you want to do.

You can also register a trigger inside a conditional block so that it is registered only if a particular condition is true at a particular time.

For example, if you always want entity 1 to respond to entry of hostile forces into an area, put a trigger at the beginning of the plan. If you want entity 1 to respond to a hostile simulation object in an area only after the simulation has progressed for 15 minutes, nest the trigger inside another trigger that fires when Sim Time > 15 minutes.

Using Triggers That Do Not Have Mutually Exclusive Tasks

A trigger does not need to have any task statements in its statement block. It could have only Set statements, or it could have sets and concurrent tasks. If a trigger block has only Set statements, it does not interrupt the task that the simulation object is executing when the trigger is executed. The Set statements are executed while the simulation object is performing its task. This has the same effect as if you manually set the simulation object's state while it was performing a task.

Using a trigger that just has Set statements can be useful if you want to change a simulation object's state under certain conditions, but do not know in advance when those conditions might exist.

2.5.2 Using Concurrent Tasks in Plans

If you are assigning tasks to simulation objects manually, you can give a simulation object a concurrent task while it is executing some other task. You cannot do this directly in a plan because you cannot assign two tasks at the same time. A plan always waits for its current task to complete before it starts the next task, even if the next task does not

conflict with the current one. If you want to assign a concurrent task in a plan, you must use a trigger. If a trigger that just has a concurrent task fires, the simulation object executes the concurrent task and continues its existing task. If the trigger has other tasks, the behavior will be as expected for triggers.

For more information about concurrent tasks, see "Concurrent Task Execution".

As an alternative to using plan triggers to automatically assign concurrent tasks, you can create reactive tasks. For information about reactive tasks, see "Reactive Tasks".

2.5.3 Moving In Formation

If you want a unit to move in a certain formation, you can set the formation (Formation set data request) or you can have the unit move into a formation (Move Into Formation task). The default formation is a column.

2.5.4 Using the Tasked-By-Superior Request in a Plan

If you put the Tasked-By-Superior request in the body of a plan, it has no real effect. This is because, as soon as it is set, the plan goes to the next task and overrides the superior. The only way that the Tasked-By-Superior command is implemented as part of a plan is if it is the last statement in a plan or the last statement in a trigger that happens to execute after all the statements in a simulation object's plan have been executed.

2.5.5 Following Simulation Objects

If you tell two or more simulation objects to follow another simulation object, do not give them the same offset values. If the followed simulation object stops, one follower will arrive at the specified offset position and stop, while the others will circle the location because they cannot all occupy the same space.

2.5.6 Using Non-VR-Forces Simulation Objects in Plans

You can use a non-VR-Forces simulation object as a parameter in a plan, for example, following a remote simulation object, or testing to see if a remote simulation object is in an area.

To identify non-VR-Forces entities, VR-Forces uses their marking text. If a non-VR-Forces simulation object does not have marking text, VR-Forces uses its simulation object identifier as its name.

VR-Forces does not recognize remote units as units, so it cannot check their children. Remote units are ignored by the simulation engine.

- Referring to remote entities by their marking text may have unpredictable results if there is more than one remote simulation object with the same marking text.
- If you include a non-VR-Forces simulation object in a plan you must ensure that the remote simulator uses the same name for the simulation object in future runs.

3. Writing Plans

This chapter explains how to combine tasks, set data requests, global commands, and conditions into plans.

3.1. Writing Plans for Simulation Objects

This section explains how to write individual plans. For conceptual information about plans, see "Introduction to Plans" on page 2.

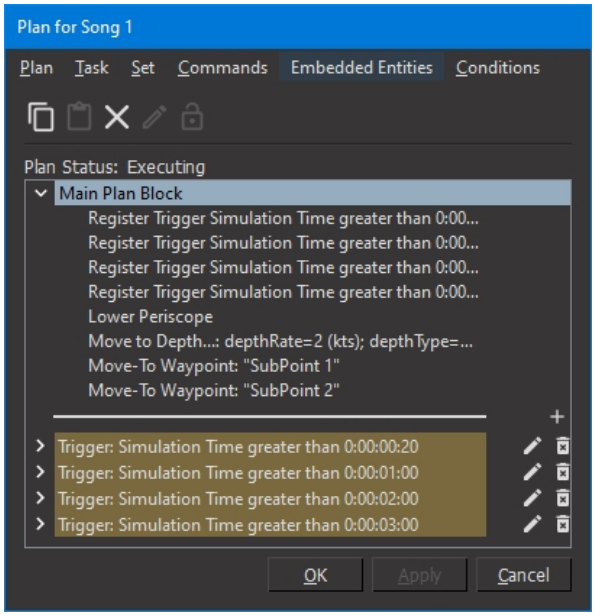
When you write a plan, keep in mind that:

- If you edit a plan during a simulation, when you apply the changes, the simulation object whose plan has been changed begins executing it from the beginning. This might mean its activities are no longer synchronized with those of the other simulation objects, unless the updated plan only includes statements that you want to execute starting at the current point of the simulation. Therefore, we recommend that you edit plans when you first load a scenario, before you start running the simulation.
- If you have two or more Plan windows open at the same time and you then open a dialog box, such as Move To, but do not complete the command, you cannot open that same dialog box from another Plan window until you complete the one that is open.

➤ **To edit a simulation object's plan:**

1. Select a simulation object.
2. Choose **Objects > Plan**, or press **P**. The Plan window opens. The simulation object's plan displays.

Figure 4. Plan window



> You can also open the Plan window by clicking the Plan button ( ) on the Last Selected Object Panel or the Command Panel.

3. Add new statements, edit statements, or delete statements as described in the following sections.

3.1.1 Adding a Task or Set Data Request to a Plan

When you add statements to a plan, you add them after the currently selected statement.

➤ **To add a task to a plan:**

1. Open the Plan window for a simulation object.
2. In the Plan window, select the statement immediately above where you want to insert a new statement. If the plan does not have any statements yet, the plan root – **Plan** is already selected.
3. Choose **Task >** *category* **>** *Task*, where *category* is one of the categories on the Task menu and *Task* is a task listed for the selected category. A dialog box appropriate to the task opens. The procedures for filling out the task parameters are described in the chapters in "Tasks".
4. Complete the required parameter information.
5. Click OK in the task dialog box to add the statement. You can click Cancel at any time to exit a task dialog box.

➤ To add a set data request to a plan, follow the same procedure as for adding a task, except choose **Set >** *category* **>** *Set Data Request*.

The procedures for filling out Set statement parameters are in the chapters in "Set Data Requests".

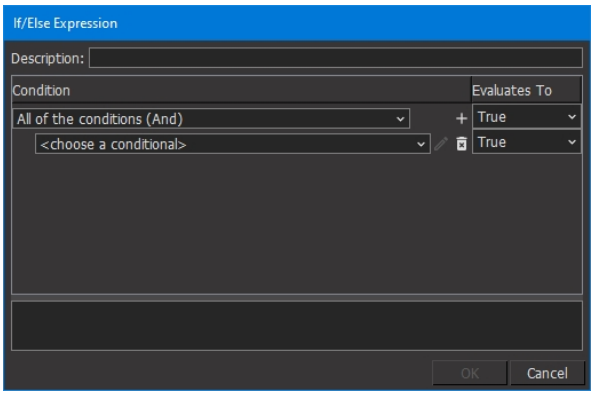
> Some task and set statements, such as Wait or Reorganize, do not require you to specify any values. They are added immediately to the plan.

3.1.2 Adding an If Block to a Plan

➤ **To add an If condition to a plan:**

1. Open the Plan window for a simulation object.
2. In the Plan window, select the statement immediately above where you want to insert the **If** block. If the plan does not have any statements yet, the plan root – **Plan** is already selected.
3. Choose **Conditions > If**. The If/Else Expression dialog box opens (Figure 5).

Figure 5. If/Else Expression dialog box



4. Optionally, in the Description box, type a description to use in the If statement instead of the condition syntax.
5. Create the conditional expression as described in "3.4. Building Condition Statements" on page 28.
6. Click OK. An **If** statement block is added to the plan.
7. Select the first line in the block (the **If** statement).
8. Add the tasks, sets, or other commands that you want to be executed if the condition is true.
9. Optionally, select the **else** line and add the commands you want to be executed if the condition is false.

3.1.3 Adding a While Block to a Plan

➤ **To add a While block to a plan:**

1. Open the Plan window for a simulation object.
2. In the Plan window, select the statement immediately above where you want to insert the **While** block. If the plan does not have any statements yet, the plan root – **Plan** is already selected.

3. Choose **Conditions > While**. The While Expression dialog box opens. (It is identical to the If/Else Expression dialog box (Figure 5).)
4. Optionally, in the Description box, type a description to use in the While statement instead of the condition syntax.
5. Create the conditional expression as described in "3.4. Building Condition Statements" on page 28.
6. Click OK. A **While** statement block is added to the plan.
7. Select the first line in the block (the **While** statement).
8. Add the tasks, sets, or other commands that you want to be executed if the condition is true.

3.1.4 Adding a Wait Until Statement to a Plan

➤ **To add a Wait Until statement to a plan:**

1. Open the Plan window for a simulation object.
2. In the Plan window, select the statement immediately above where you want to insert the **Wait Until** statement. If the plan does not have any statements yet, the plan root – **Plan** is already selected.
3. Choose **Conditions > Wait Until**. The Wait Until Expression dialog box opens. (It is identical to the If/Else Expression dialog box (Figure 5).)
4. Optionally, in the Description box, type a description to use in the **Wait Until** statement instead of the condition syntax.
5. Create the conditional expression as described in "3.4. Building Condition Statements" on page 28.
6. Click OK. A **Wait Until** statement is added to the plan.

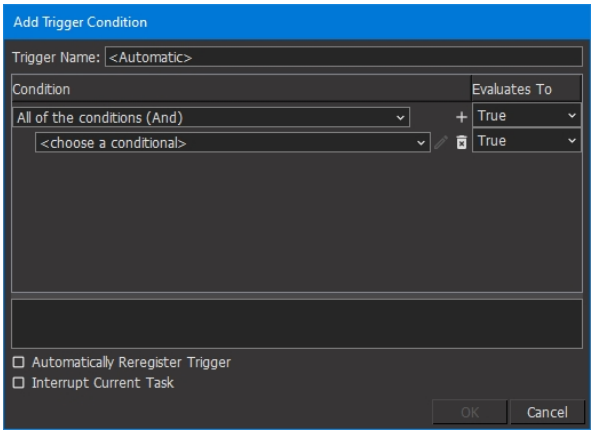
3.1.5 Adding a Trigger to a Plan

Adding a trigger to a plan is a three-step process. First, you specify the trigger condition. Then you add the commands to be executed to the trigger body. Finally, you register the trigger in the body of the plan.

➤ **To add a Trigger to a plan:**

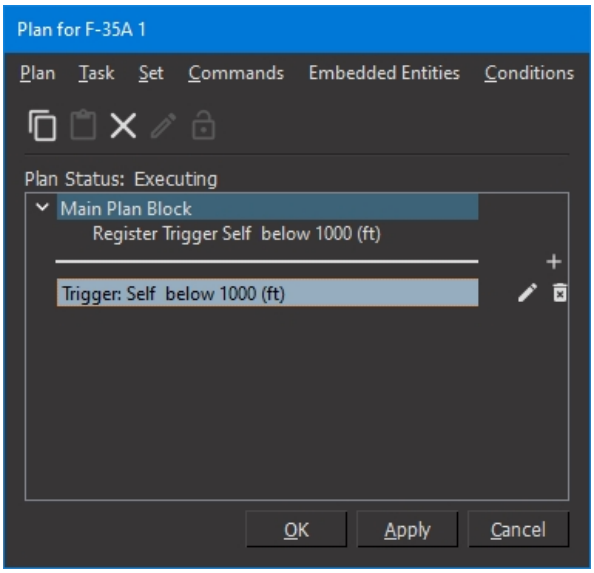
1. Open the Plan window for a simulation object.
2. Choose **Conditions > Add Trigger**. The Edit Trigger Condition dialog box opens (Figure 6). (You can also add a trigger by choosing **Conditions > Register Trigger** and selecting New in the trigger list.)

Figure 6. Trigger dialog box



3. Optionally, in the Trigger Name box, enter a name for the trigger. If you specify a name, it displays in the Register command when you register the trigger. If you do not specify a name, the text of the condition displays in the Register command.
4. Optionally, select the Automatically Reregister Trigger check box to specify that the trigger should be automatically reregistered after it runs.
5. Optionally, select the Interrupt Current Task check box to specify that the trigger should interrupt the currently running task rather than suspending and resuming it.
6. Create the conditional expression as described in "3.4. Building Condition Statements" on page 28.
7. Click OK. The trigger name is listed below the main plan block. (Figure 7).

Figure 7. Trigger pane



8. Select the first line in the Trigger pane (**Trigger Body**).

9. Add the tasks, sets, or other commands that you want to be executed when the condition becomes true.
10. Register the trigger. For details, see "3.1.6 Registering Triggers" below.

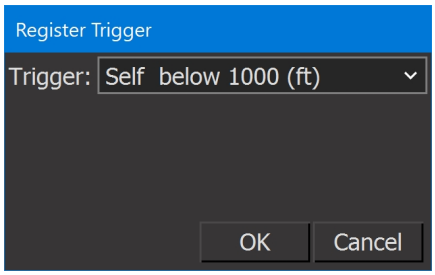
3.1.6 Registering Triggers

Writing a trigger does not necessarily include it in a plan. To use a trigger, you must register it. If you create a new trigger using **Conditions > Add Trigger**, you must register it manually. If you create a new trigger by choosing **Conditions > Register Trigger** and selecting New, it is automatically registered at the currently selected location in the plan.

➤ **To register a trigger:**

1. In the Plan window, select the statement below which you want to register the trigger.
2. Choose **Conditions > Register Trigger**. The Register Trigger dialog box opens (Figure 8).

Figure 8. Register Trigger dialog box



3. Select the trigger you want to register from the list. (If there are no triggers in the plan, or you want to create a new one, select New.)
4. Click OK.

Reregistering Triggers

By default, once the statements in a trigger are executed, the trigger is unregistered and VR-Forces no longer tests the condition. However there may be cases where you want to test for a recurring condition. You could register the trigger again when the trigger block competes. Or, when you create or edit a trigger, you can specify that it automatically reregister. For details, see "3.1.5 Adding a Trigger to a Plan" on page 23.

As an alternative to using triggers, you can write reactive tasks, which work similarly to triggers, or you can write a regular scripted task that is designed to repeatedly test a condition.

Unregistering Triggers

You can unregister triggers from within a plan. Unregistering a trigger does not remove it from a plan. It causes VR-Forces to stop testing the condition. You might want to unregister a trigger in response to a change in scenario circumstances that makes the actions in the trigger undesirable.

➤ **To unregister a trigger:**

1. Select the plan statement above where you want to add an Unregister Trigger statement.
2. In the Plan window, choose **Conditions > Unregister Trigger**. The Unregister Trigger dialog box opens.
3. Select the trigger that you want to unregister.
4. Click OK.

3.1.7 Cleaning Up Unused Triggers

If you add triggers to a plan and then decide not to use (register) them, you can quickly remove them from the plan. (However, there is no penalty for leaving unused triggers in a plan.)

➤ To clean up unused triggers, choose **Plan > Clean Up Unused Triggers**.

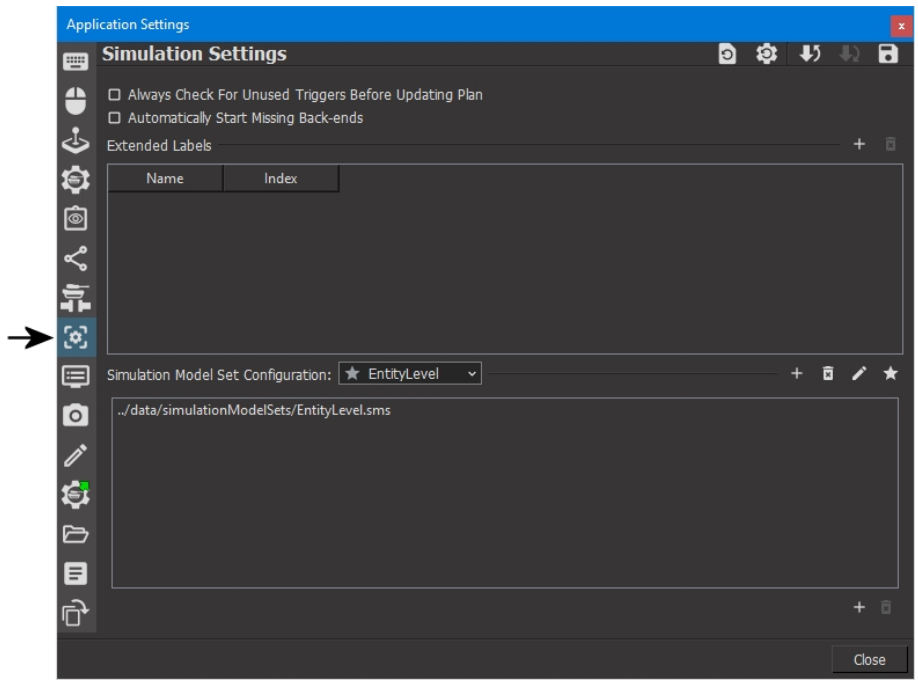
Automatically Checking for Unused Triggers

VR-Forces can automatically check for unused triggers when you save or apply changes to a plan. If it finds any, it displays a dialog box offering you options for handling them.

➤ **To specify that VR-Forces should check for unused triggers when it updates a plan:**

1. Choose **Settings > Application**. The Application Settings dialog box opens.
2. Select the Simulation Settings page (Figure 9).

Figure 9. Simulation Settings page



3. Select the Always Check For Unused Triggers Before Updating Plan check box.
4. Click Close.

3.1.8 Editing a Statement

You can edit simple statements in the Plan window.

➤ **To edit a plan statement:**

1. Double-click the statement or select the statement and then click the Edit button (). An edit dialog box for the particular statement type opens.
2. Enter the changes you want to make.
3. Click OK in the edit dialog box.
4. Click OK or Apply in the Plan window.

3.1.9 Deleting Statements from a Plan

You can delete statements from a plan individually, or all at once. You can delete the individual statements in the statement block of a conditional statement. If you delete the conditional statement itself, all substatements are also deleted.

➤ **To delete one statement from a plan:**

1. Select the statement you want to delete.
2. Click the Delete button () or press **Delete**.

➤ **To delete all statements in a plan:**

1. Select the top line in the plan.
2. Click Delete () or press **Delete**.

3.2. Printing Plans

You can print plans. The printout lists the name of the scenario, the simulation object's name, and the plan statements.

➤ **To print a plan:**

1. Open a global plan or the Plan window for the simulation object whose plan you want to print.
2. Choose **Plan > Print**. A standard print dialog box for your operating system opens.
3. Complete the dialog box as you normally would.

3.3. Saving Changes to a Plan

You can save changes to a plan and continue editing, or save changes and exit the Plan window. The plan is saved to the scenario plan file. You do not specify a file to save it to.

➤ To save changes to a plan, in the Plan window, click Apply or OK.

3.4. Building Condition Statements

If you want to add an **If** block, a **While** loop, a **Wait Until**, or a trigger, you must specify a condition to be tested. The conditional expression dialog box is a dynamic dialog box that lets you build a condition. It contains lists that have the permissible options for each step of a condition. As you select options, the dialog box reloads, adding additional fields. You can build condition statements using comparison operators, logical operators, and tests of simulation object status.

A condition statement can have one condition or many. They can be nested and grouped. In the following sections, we explain how to build a simple condition and how to build a complex condition.

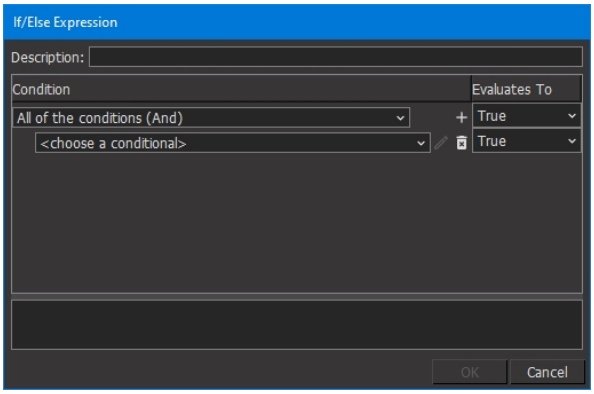
3.4.1 Building a Condition Statement with One Condition

Although the If/Else Expression, While Expression, Wait Until Expression, and Trigger dialog boxes differ slightly, the process of building a condition is the same for all of them.

➤ **To build a condition statement that has one condition:**

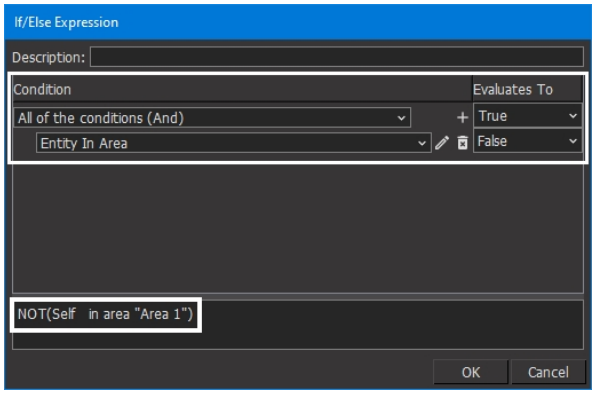
1. On the Conditions menu, choose the type of condition you want to create. The applicable dialog box opens (Figure 10). The top line in the condition specifies that all of the conditions are True. Since we are creating a condition statement that just has one condition, this line is largely irrelevant except for the Evaluates To column.

Figure 10. If/Else Expression dialog box



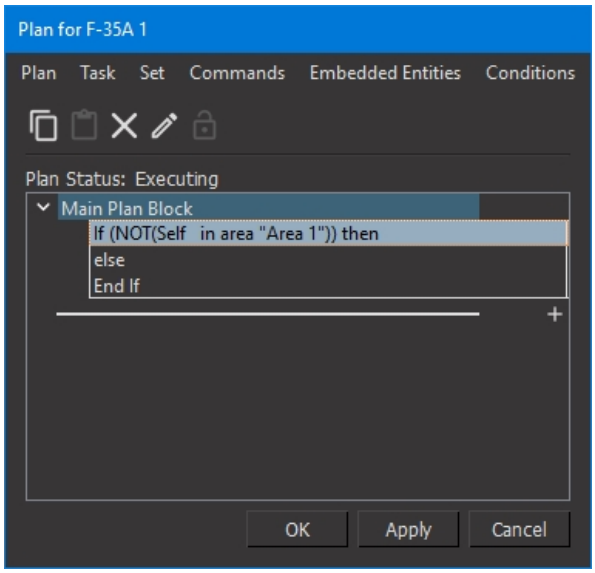
2. Optionally, in the Description box, type a short summary of the condition.
3. Optionally, in the Evaluates To list, change the entire condition from True to False.
4. In the <choose a conditional> list, select the condition that you want to test. If applicable, a dialog box opens for you to specify the parameters of the condition.
5. Specify the parameters of the condition.
6. Click OK. The condition is displayed in the display box at the bottom of the dialog box (Figure 11).
7. Optionally, in the Evaluates To list, change the condition from True to False. Figure 11 shows a simple condition.

Figure 11. Simple conditional expression



8. Click OK. The condition statement is added to the plan or the Trigger pane. In Figure 12, note that the If statement uses the text from the description (in Figure 11) instead of the full syntax of the condition shown at the bottom of the If/Else Expression dialog box.

Figure 12. Simple conditional expression in a plan



3.4.2 Building a Complex Condition

You can build condition statements with multiple conditions, nested conditions, and logical operators. The condition expression dialog box displays the syntax of the condition as you build it. The layout of the window shows which conditions are grouped and which are subordinate to others. For an example of building a complex condition, see "Complex Condition Tutorial" on the next page.

➤ **To build a complex condition:**

1. On the Conditions menu, choose the type of condition you want to create. The applicable condition expression dialog box opens. The top line in the condition specifies that all of the conditions are True.
2. Optionally change the top level logical operator to Or.
3. Optionally change the top level evaluator to False.
4. In the <choose a conditional> list, select the condition that you want to test. If applicable, a dialog box opens for you to specify the parameters of the condition.
5. Specify the parameters of the condition.
6. Click OK. The condition displays in the display box at the bottom of the dialog box.
7. Click the Add button (+) next to the top line of the condition. A new condition line is added.
8. Select a condition to test and specify its parameters.
9. Continue to add conditions as desired. The condition syntax updates as you add conditions.
10. Optionally, select a new logical operator in the conditions list (And or Or). A subordinate condition line is added under the logical operator and indented.
11. Add conditions for the logical operator group as you did for the top level logical operator. To simplify the view, you can expand and collapse the logical grouping by clicking the arrow to the left of the logical operator line.
12. Continue adding conditions as desired.
13. Click OK.

Complex Condition Tutorial

This section walks you through the process of creating a complex condition. It creates the following conditional expression:

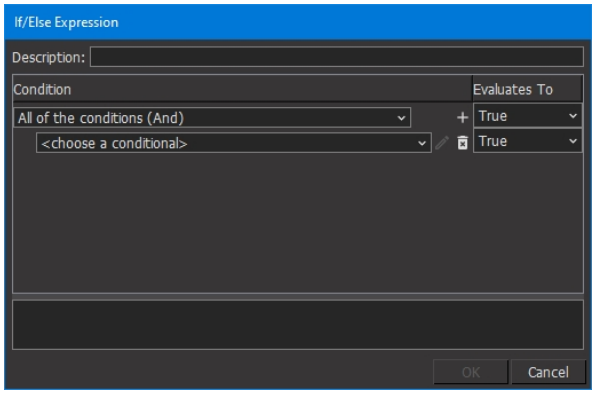
```
  If("M1A2 4" in area "Area 1" AND ("M1A2 2" left of "Phase Line 1" OR
Entity "T80 1" destroyed))then
  else
  endif
```

The procedure assumes a scenario with four M1A2 entities, a T-80, a phase line, and an area.

➤ **To build the condition statement:**

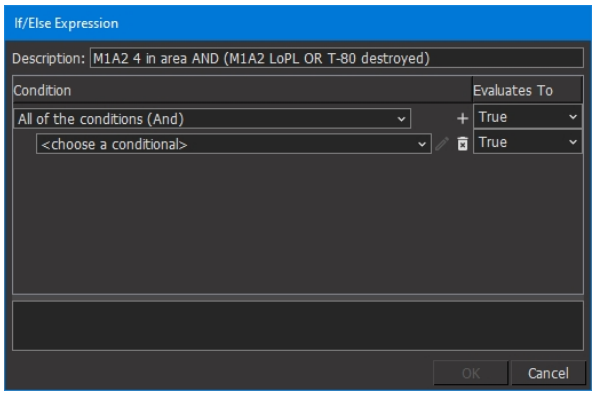
1. Open the Plan window for M1A2 1.
2. Choose **Conditions > If**. The If/Else Expression dialog box opens (Figure 13).

Figure 13. If/Else Expression dialog box



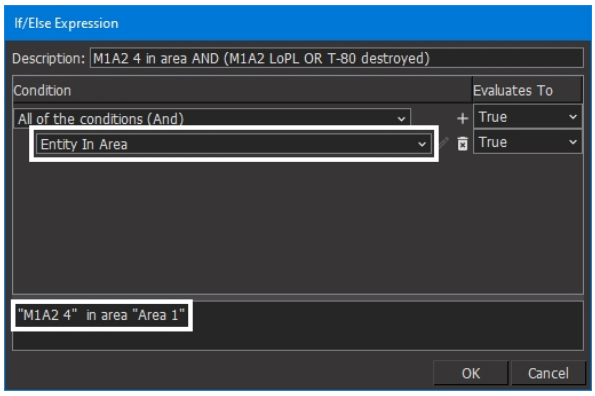
3. Type a description for the expression you want to build (Figure 14).

Figure 14. Description added



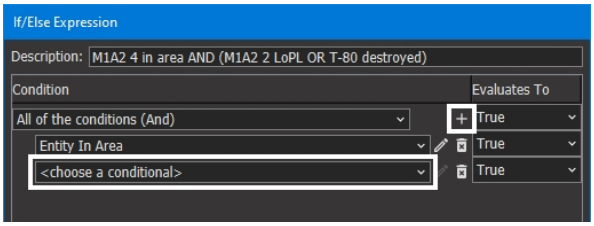
4. In the <choose a conditional> list, select **Entity in Area**. The Entity in Area dialog box opens.
5. In the Entity in Area dialog box, select M1A2 4 and Area 1.
6. Click OK. The condition is added and the If expression is updated (Figure 15).

Figure 15. Entity in Area condition added



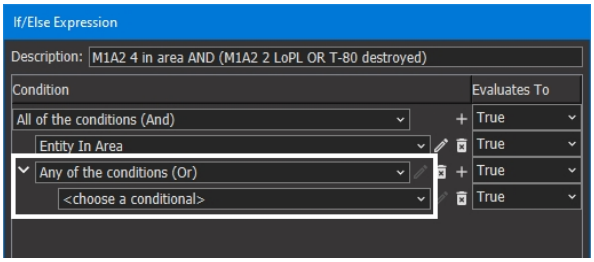
7. On the All of the conditions line, click the Add button (+). A new line is added to the window (Figure 16).

Figure 16. Condition added



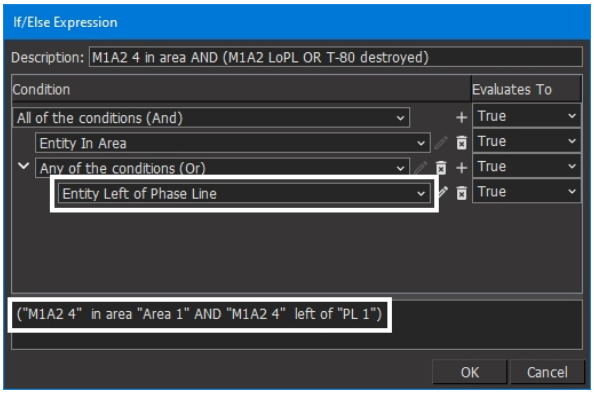
8. In the <choose a conditional> list, select **Any of the conditions (Or)**. A new line is added to the window (Figure 17).

Figure 17. Or operator and condition added



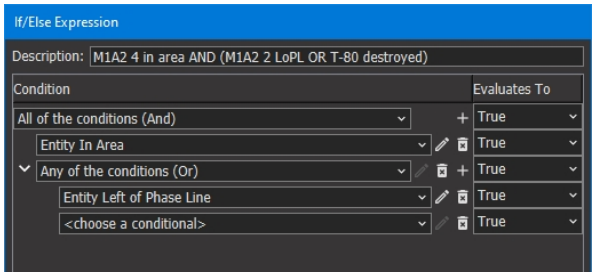
9. In the <choose a conditional> list, select **Entity Left of Phase Line**. The Entity Left of Phase Line dialog box opens.
10. Select M1A2 2 and Phase Line 1.
11. Click OK. The condition is added and the If expression is updated (Figure 18).

Figure 18. Entity Left of Phase Line condition specified



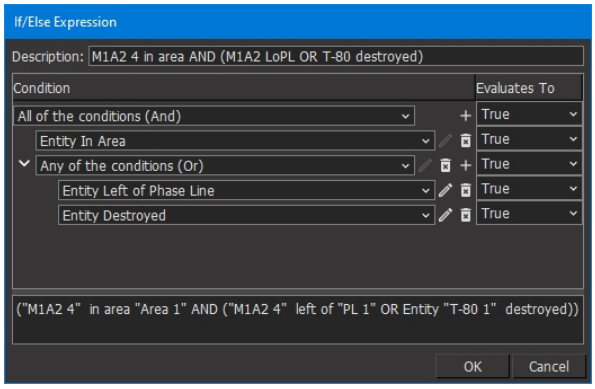
12. On the Any of the conditions (Or) line, click the Add button. A new line is added to the window (Figure 19).

Figure 19. Last condition line added



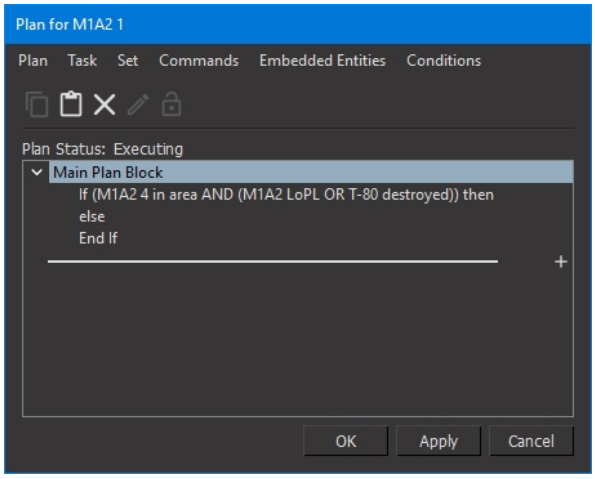
13. In the <choose a conditional> list, select **Entity Destroyed**. The Entity Destroyed dialog box opens.
14. Select T-80 1.
15. Click OK. The condition is added and the If expression is updated (Figure 20).

Figure 20. Completed expression



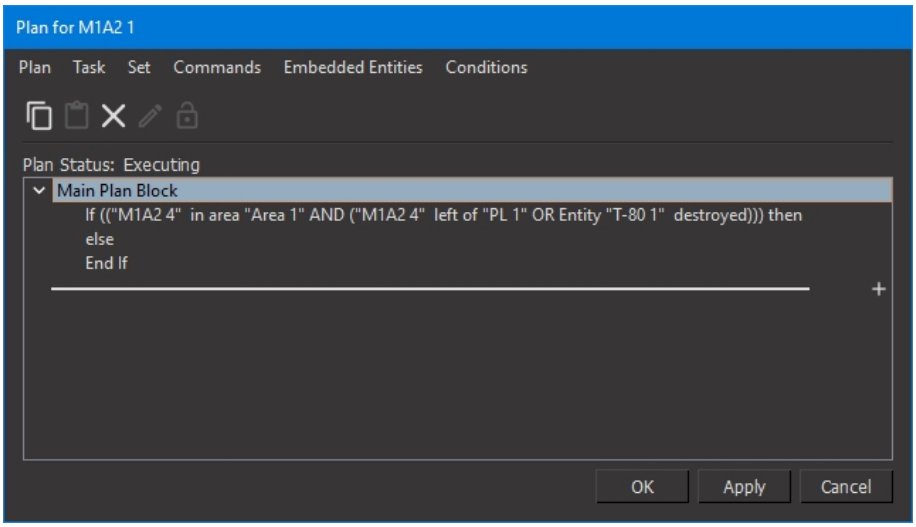
16. In the If/Else Expression dialog box, click OK. The If expression is added to the plan window. Notice that the expression uses the text from the Description box (Figure 21).

Figure 21. If expression using condition description



17. If you did not include a description, the entire syntax of the expression displays (Figure 22).

Figure 22. If expression using expression syntax



3.4.3 Editing Condition Statements

➤ **To edit an If, While, or Wait Until condition statement:**

1. In the Plan window, double-click the statement. The applicable expression building dialog box opens.
2. To edit the parameters of a condition, click the Edit button (✎) for that condition and change the parameters as desired.
3. To change the condition being tested, select a different condition in the list and specify its parameters.
4. To delete a condition, click the Delete button (🗑) for that condition.
5. To change how the condition is evaluated, choose a different option in the Evaluates To list.
6. Click OK.

➤ **To edit a Trigger:**

1. In the Triggers list, found below the Main Plan Block, select the trigger that you want to edit.
2. Click the Edit button (✎). The Trigger dialog box opens.
3. To edit the parameters of a condition, click the Edit button (✎) for that condition and change the parameters as desired.
4. To change the condition being tested, select a different condition in the list and specify its parameters.
5. To delete a condition, click the Delete button (🗑) for that condition.

6. To change how the condition is evaluated, choose a different option in the Evaluates To list.
7. Click OK.

3.4.4 Specifying Names or Patterns in Condition Statements

Many of the conditions that you can use to build a condition statement require you to specify simulation objects for which the condition will be tested. You can specify:

- A specific named simulation object, for example, simulation object M1A2 1 is in Area 1.
- That the condition applies to "Self", for example, "I am in Area 1". Specifying a condition for one's self makes it easy to copy plans or plan statements and paste them into the plans of other simulation objects without doing any editing.
- An enumeration, for example, any simulation object with enumeration 1:1:225:1:1:3:0 (any M1A2).
- The inverse of the selection, for example, any simulation object but myself, or any simulation object except an M1A2.
- That for units, the entire unit must satisfy the condition (the default), or that any subordinate member of a unit satisfies the condition. For example, the default behavior when Entity Is Destroyed is applied to a unit is that all members of the unit must be destroyed for the condition to be true. The alternative is that the condition is true if any member of the unit is destroyed.

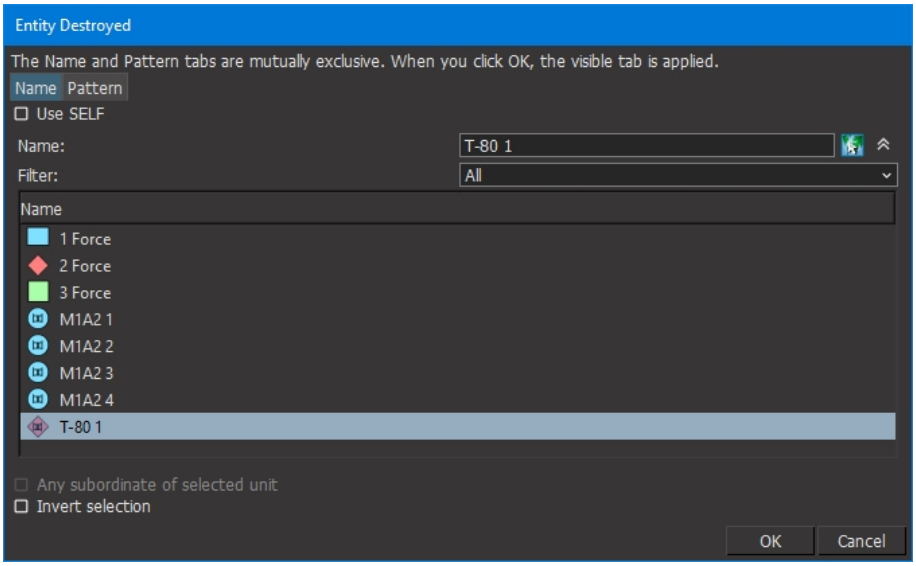
The Name and Pattern specifications are mutually exclusive. You cannot specify a name and a pattern. The tab in the Condition dialog box that is visible when you click OK is the specification that is applied.

Specifying a Name in a Condition Statement

➤ **To specify a named simulation object in a condition statement:**

1. In a condition dialog box, select the Name tab (Figure 23).

Figure 23. Name tab in condition dialog box



2. To specify the simulation object whose plan this is as the simulation object in the condition, select the Use SELF check box.

   To specify a simulation object by name:

   a. Optionally filter the list. If you want to use a plan variable, you must select the Plan Variables filter.

   b. Type the name of the simulation object or variable or expand the list and then select it.

3. If you select a unit in the list, the Any Subordinate of Selected Unit check box is enabled. If you want the condition to be satisfied when any member of the unit meets the condition, select the check box.

4. To specify the inverse of the name, that is, any member except the selected one, select the Invert Selection check box.

   ⓘ Selecting both Any Subordinate of Selected Unit and Invert Selection causes the condition to be true if it is true for any entity in the scenario.

5. Click OK.

Specifying a Pattern in a Condition Statement

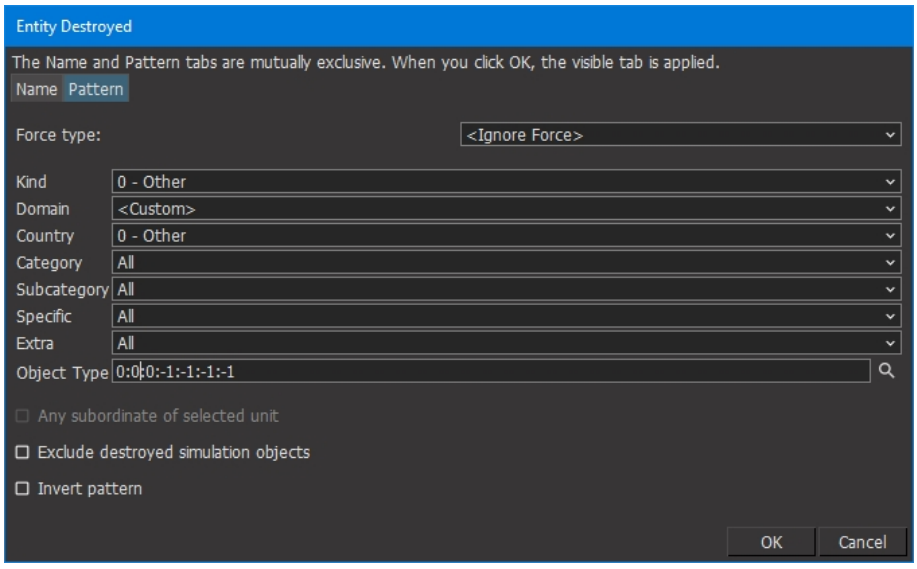
The enumerations used in the lists on the Pattern tab are loaded from the file *./appData/settings/vrfGui/condition-entity-types.csv*. They represent a subset of the enumerations typically used for simulation objects in simulations. However, you can enter any numeric value you want in the enumeration value box.

If you expect that you will consistently need to specify enumeration values that are not part of *condition-entity-types.csv*, such as a particular country code, you can edit it to add the patterns you need.

➤ **To specify a pattern in a condition statement:**

1. In a conditional statement dialog box, select the Pattern tab (Figure 24).

Figure 24. Pattern tab in condition dialog box



2. Optionally, specify a force to test for.
3. For each element of the enumeration, select an option from the list, or type an enumeration in the text box.
4. If the enumeration you enter is for a unit, the Any Subordinate Of Selected Unit check box is enabled. If you want the condition to be satisfied when any member of the unit meets the condition, select the check box.
5. To specify the inverse of the name, that is, any member except the selected one, select the Invert Pattern check box.
6. Click OK.

3.5. Using Plan Variables

You can use plan variables in most places where you would select a specific simulation object in a task, set, or conditional test. Plan variables use the names Simulation Object and Created Object. You cannot add your own variables or change the names.

The variables are assigned values only after an object has been created from within a plan (Created Object) or a conditional test has been met that identifies a simulation object (Simulation Object). If you use a variable in a plan that does not have a value, it is ignored.

A Simulation Object variable is not assigned a value until a condition has been completely evaluated. Therefore, you cannot do something like If Entity In Area AND Simulation

Object is Destroyed and expect that the Simulation Object variable refers to the entity identified by Entity in Area. In this case, the Simulation Object variable would refer to a previously identified simulation object or it would just be empty.

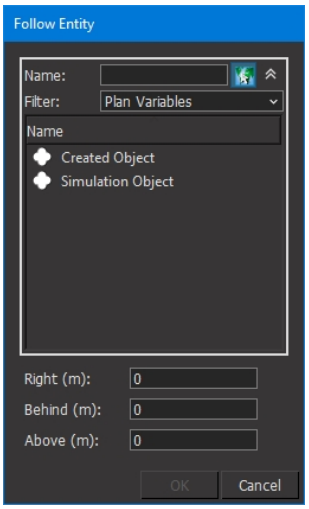
> ⓘ Remember that the values assigned to plan variables can change as a simulation object progresses through its plan.

For conceptual information about plan variables, see "2.3. Plan Variables" on page 14.

➤ **To use a plan variable in plan statement:**

1. In the statement creation dialog box, select Plan Variables in the Filter list. The dialog box lists Created Object and Simulation Object (Figure 25).

Figure 25. Plan variables



2. Select the variable that you want to use.
3. Complete the rest of the dialog box normally.

3.6. Creating Global Plans

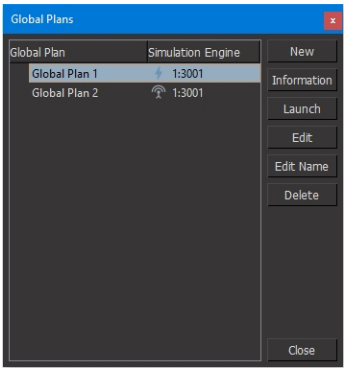
Global plans are not tied to a specific simulation object. However, since they must be executed by a simulation engine, they are assigned to a specific sim. engine.

Global plans have a limited set of Task and Set commands and the same set of Commands menu options and conditions as individual plans. (You can add additional tasks using the VR-Forces API or by writing scripted tasks.) You can open an Information dialog box for a global plan to see its task status and console messages.

➤ **To create a global plan:**

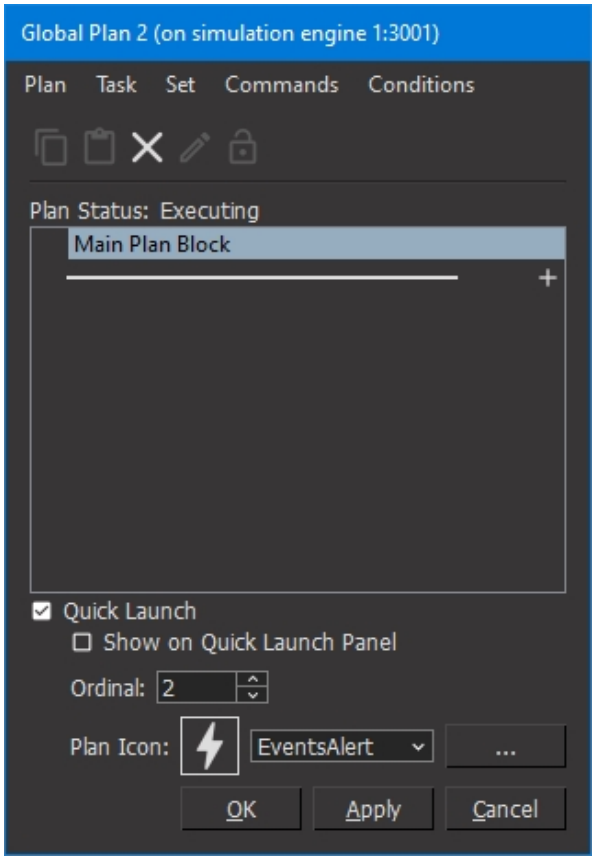
1. Choose **Simulation > Global Plans** or click the Global Plans button ( ) in the Scenario Components Toolbar. The Global Plans window opens (Figure 26). It lists each global plan and the simulation engine on which it is executed.

Figure 26. Global Plans window



2. Click New. The New Global Plan Information dialog box opens.
3. In the Plan Name box, enter a name for the plan.
4. If your scenario has more than one sim. engine, select the sim. engine on which you want to run the global plan.
5. Click OK. The Global Plan window opens (Figure 27).

Figure 27. Edit Global Plan window



6. Add commands from the Task, Set, or Conditions menu, as described in "3.1. Writing Plans for Simulation Objects" on page 20. Add commands from the Commands menu as described in "3.7. Adding Commands to a Plan" on the facing page.
7. Optionally, select the Quick Launch check box to enable quick launching of this plan. For information about using the Quick Launch Panel, see "3.6.1 Controlling When Global Plans Run (Quick Launch)" below.
8. Optionally, select the Show on Quick Launch Panel check box to add this global plan to the Quick Launch Panel.
9. Optionally, in the Ordinal box, specify the position of this plan on the Quick Launch Panel.
10. Optionally, change the icon for this plan on the Quick Launch Panel:
    a. Click the Browse button next to the icon name. A file chooser window opens.
    b. Select the file you want to use for the icon.
    c. Click Open.
11. Click OK.

3.6.1 Controlling When Global Plans Run (Quick Launch)

If you enable quick launch for a global plan, it does not start executing until you launch it. You can launch it from the Quick Launch Panel, from the Global Plans window, or from another plan. (You can enable quick launch without putting an icon on the Quick Launch Panel.) When you launch a global plan, it runs to the end. When it finishes, you can launch it again.

> ⓘ The Quick Launch Panel shows an icon and text for each quick launch item. The text is the name of the plan.

➤ To launch a global plan from the Quick Launch Panel, click the icon and text.

➤ **To launch a global plan from the Global Plans window:**

1. Choose **Simulation > Global Plans** or click the Global Plans button () in the Scenario Components Toolbar. The Global Plans window opens (Figure 26).
2. Select the global plan you want to launch. If it is not running, the Launch button becomes active.
3. Click Launch. The global plan runs. When it completes, the Launch button becomes active again.

➤ **To launch a quick launch global plan from a plan:**

1. Open the plan window for an individual plan or a global plan.
2. Choose **Commands > Launch Global Plan**. The Launch Global Plan window opens.
3. Select the quick launch global plan you want to launch.
4. Click OK.

3.6.2 Opening an Information Dialog Box for a Global Plan

➤ **To open an Information dialog box for a global plan:**

1. Choose **Simulation > Global Plans** or click the Global Plans button ( ) in the Scenario Components Toolbar. The Global Plans window opens (Figure 26).
2. Select the global plan for which you want to open an Information dialog box.
3. Click Information.

3.7. Adding Commands to a Plan

The Commands menu lets you assign tasks and set data requests to simulation objects and create the objects on the Create Object Palette. There are also several commands that are only available on the Commands menu. Except as noted in this section, the procedures for adding global commands and conditional statements to a global plan are the same as the procedures for adding them to individual plans.

Since a global plan has no idea what type of simulation object you might want to give a task or set data request to, when you add a task or set, the full list of possible tasks and sets is available. There is no context sensitivity like there is when you select a simulation object and view the Task or Set menu. It is up to you to be sure that the task or set you are assigning to a simulation object is supported by that object.

The best way to determine this is to create a simulation object of the type to which you want to give a task and see if the task or set data request is available on its context-sensitive Task or Set menu. You can also check the Simulation Object Editor to see if a simulation object has the system used for a particular task.

3.7.1 Adding Tasks and Sets from the Commands Menu

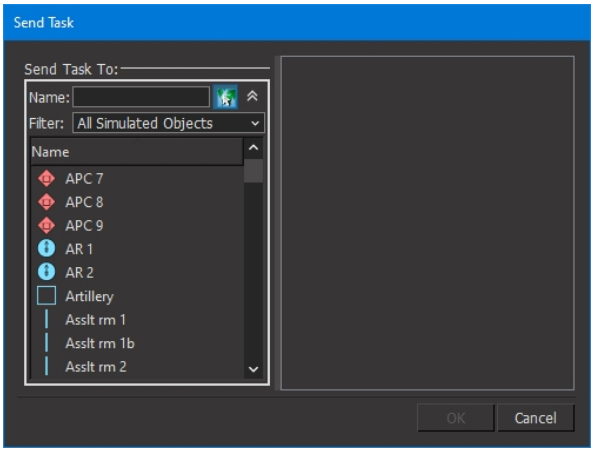
The commands on the Task and Set menus in an individual plan apply implicitly to the simulation object for whom you are writing a plan. Tasks and sets on the Commands menu must be assigned explicitly to a target simulation object.

Other than the requirement to specify the target simulation object, adding a global task or set command to a plan uses the same procedure as adding any task or set to an individual plan.

➤ **To assign a task from the Commands menu:**

1. In a Plan window or Global Plan window, choose **Commands > Send Task**. The Send Task dialog box opens (Figure 28).

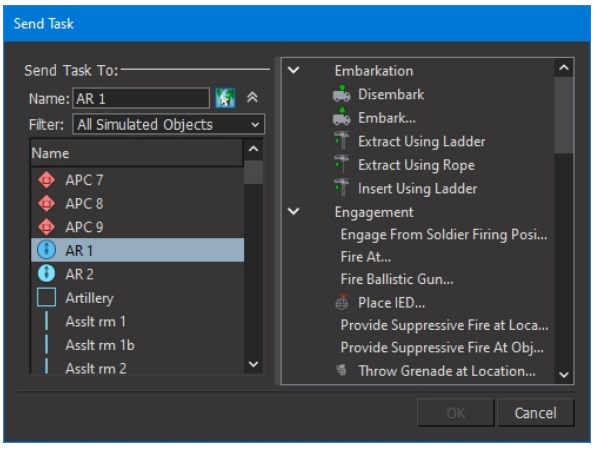
Figure 28. Task dialog box



> The procedure for adding a set data request is the same as for a task, except that you choose **Commands > Send Set** and the Send Set dialog box opens.

2. In the Apply Command To group box, select a simulation object from the list or type its name in the Name box. The simulation object does not have to exist in the scenario at the time you add the command, but you should be sure that it will be created before the command is run. The window displays a list of tasks for the selected simulation object (Figure 29). If the simulation object already exists in the scenario, the list of tasks is context-sensitive.

Figure 29. Task list



> Remember that a command task affects a simulation object like an independent task. It interrupts the current task and causes the simulation object to abandon its plan. If you are adding a command to an individual plan, be sure that you are assigning it to some other simulation object and not the simulation object for whom you are writing the plan. (Unless you want the simulation object to abandon its plan.)

3. Select the task you want to assign.
4. Click OK. If the task takes parameters, a dialog box opens.

   If the task does not take parameters, such as Wait, the command is added to the plan.
5. If the task takes parameters, specify the parameters and click OK in the task-specific dialog box. See the chapters in "Tasks" and "Set Data Requests" for the details of assigning specific tasks and sets.
6. Click OK. The command is added to the plan.

Editing Task and Set Commands in Plans

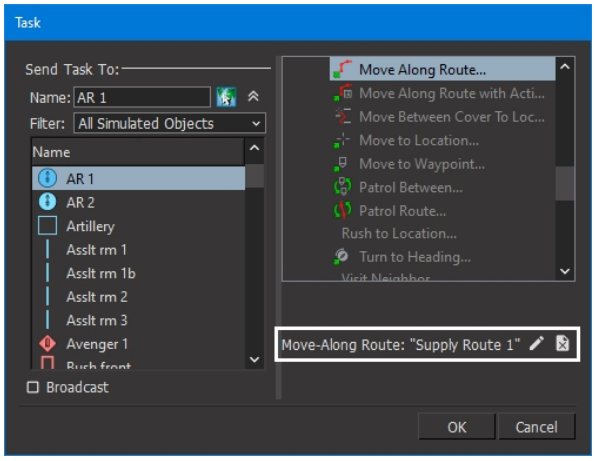
If you want to edit a task or set command in a plan, you can:

- Delete the plan statement and start over.
- Edit the existing statement. If you edit the existing statement, you can:
  - Delete the current task and choose a different one.
  - Edit the parameters of the task.

➤ **To edit a task or set command in a plan:**

1. Select the plan statement you want to edit.
2. Click the Edit button, or double-click the statement. The Send Task or Send Set dialog box opens. The task list is grayed out. If you are not sure of the task that you are editing, you can read the text of the command below the window (Figure 30).

Figure 30. Send Task dialog box for plan command



3. To keep the same task and change the parameters, click the Edit button (). The appropriate dialog box opens and you can edit the parameters.

   To clear the task and start over, click the Delete button ().
4. Click OK.

3.7.2 Sending Text Messages from the Commands Menu

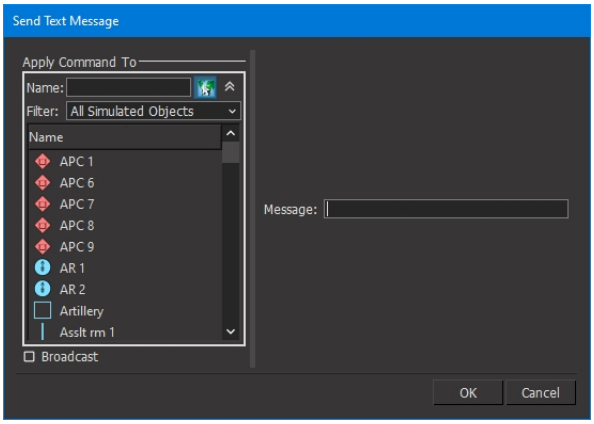
The Send Text Message command lets you send a text message to a simulation object without using the radio network. The receiving simulation object can test for receipt of the message in a conditional statement in its plan. (For details, see "Receive Text Message" on page 13.)

Because the Send Text Message command does not use the radio network, simulation objects in forces other than the sending simulation object can receive the text message. Additionally, text messages sent using the Send Text Message command do not display communication lines and the sending object does not display squawk indicators.

➤ **To send a text message from the Commands menu:**

1. In a Plan window or Global Plan window, choose **Commands > Send Text Message**. The Send Text Message dialog box opens (Figure 31).

Figure 31. Send Text Message dialog box for plan command



2. To send the message to all simulation objects in the scenario, select the Broadcast check box.

   To send the message to a specific simulation object, in the Name text box, type the name of the simulation object that will receive the message, or select the simulation object in the list or on the terrain.

3. In the Message box, type the text message.
4. Click OK. The command is added to the plan.

3.7.3 Sending Console Messages

You can send messages to be displayed in the console on an object's Information dialog box and to the sim. engine console. Messages sent by global plans go to the sim. engine console. Messages sent from a simulation object's plan go to the console in its Information dialog box.

If the notification level for a console message is lower than the notification level configured for the sim. engine or the target simulation object, the message is not displayed. For example, if a simulation object's notification level is set to Info, it receives Error, Warn, and Info messages, but not messages set to Verbose or Debug.

➤ **To send a console message from a plan:**

1. In a plan window, choose **Commands > Console Message**. The Console Message dialog box opens.
2. Select the notify level from the list.
3. Type the message you want to send.
4. Click OK.

3.7.4 Creating Objects from Within a Plan

You can create any of the objects on the Create Object Palette from within a plan.

➤ **To create an object from a plan:**

1. In a plan window, choose **Commands > Create >** *palette*, where *palette* is one of the object creation palettes. The selected palette opens.
2. Select the object you want to create.
3. Create the desired object as you would if you were creating it directly from one of the object creation palettes. When you are finished creating the object, a Create command is added to the plan and the object itself disappears. (The object disappears because you are not actually creating the object right now, you are just creating the command that will create it when the plan executes.)

3.7.5 Deleting Objects from Within a Plan

You can delete any object in a scenario from a plan.

➤ **To delete an object from a plan:**

1. In a plan window, choose **Commands > Delete Object**. The Delete Object dialog box opens.
2. Type the name of the object you want to delete or select the object in the list or on the terrain.
3. Click OK.

Deleting One's Self from the Scenario

The Delete Object global command lets you delete any specified object. If you just want to delete the simulation object whose plan you are editing, you can use the Delete Self command. Like the Use Self option in some conditions, Delete Self provides a non-entity-specific command that is useful for plans that you want to assign to multiple simulation objects.

➤ To assign the Delete Self command, choose **Commands > Delete Self**.

3.7.6 Issuing a Plan

The Issue Plan global command lets you assign a plan to a simulation object. When you issue a plan, the new plan replaces whatever plan or task the simulation object is executing. Although you can issue a plan to any simulation object, this command is particularly useful when you create a simulation object after a scenario has started.

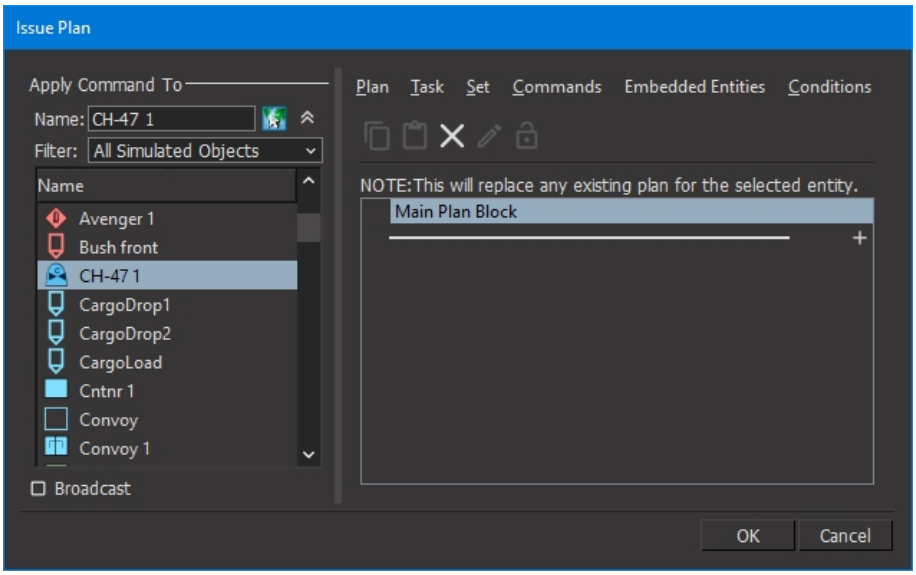
When you create a scenario, you cannot create a plan for a simulation object that does not exist. Therefore, if you create a simulation object after the scenario starts (using the Create global command), it is difficult to send it a sequence of tasks because global commands are not executed in sequence (as discussed in "2.1. Introduction to Individual Plans and Global Plans" on page 4). The Issue Plan command solves this problem.

Issue Plan is a concurrent task. Including it in a plan does not affect the current task that the simulation object is executing or the sequencing of tasks in its plan.

➤ **To issue a plan to a simulation object:**

1. In a plan window, choose **Commands > Issue Plan**. The Issue Plan dialog box opens (Figure 32).

Figure 32. Issue Plan dialog box



2. In the Apply command To: group box, select the simulation object to which you want to assign this plan or type the name of the simulation object in the Name text box.

   If this is a simulation object that does not yet exist (because you plan to create it after the scenario starts), you must type the name and you are responsible for ensuring that the name you type matches the name of the simulation object that will be created. You must also ensure that the scenario does not have more than one entity with the same name.

   If you want to issue a plan to an entity that has just been created, you can use the Created Object plan variable to identify the object. For more information about the Created Object variable, see "2.3. Plan Variables" on page 14.

3. Use the right side of the dialog box to write a plan. It has the exact same set of menus and commands as the Plan dialog box. Follow the procedures described elsewhere in this chapter to write the plan.

4. Click OK.

3.7.7 Sending View Control Messages

You can send view control messages to set the view mode for observers in the simulation. This includes the observers in VR-Forces and in any other VR-Vantage applications in the exercise. (The VR-Forces GUI is built using the VR-Vantage toolkit and is, therefore, considered a VR-Vantage application.) When you send a command you can direct it to a specific observer or to all similarly named observers, for example, all Observer 1s.

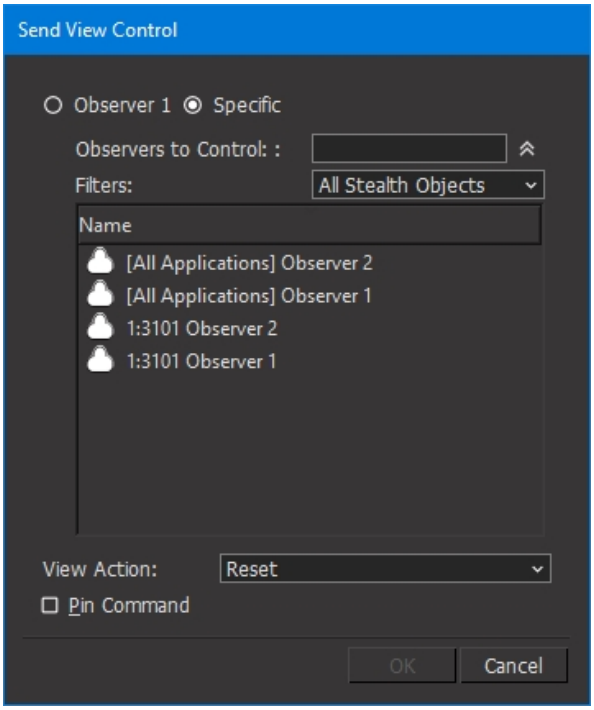
You can also send saved observer views. Observer views work only with VR-Forces. They do not affect other VR-Vantage applications. If you send an observer view, you can specify a transition time to the new view, similar to the automated transition described in "Animating Transitions Between Observer Views".

- The View Control command does not support all of the view modes. However, you can create a saved view using any view mode and send the saved view (to VR-Forces only).
- If you send an observer view, the current values for the observer view are put into the plan. If you subsequently change the observer view, the changed values are not picked up.
- If you record a scenario with the MAK Data Logger, it records view control commands. You can use them to create demonstration recordings that automatically change to different views as the recording progresses.

➤ **To send a view control message:**

1. In a plan window, choose **Commands > Send View Control**. The Send View Control window opens (Figure 33).
2. Select an Observers to Control option. Observer 1 sends the view control command to Observer 1 in all VR-Vantage applications. If you want to direct the view control message to a particular observer:
    a. Select Specific. The dialog box expands to show a list of observers.
    b. Select the observers you want to control by object name (such as 1:3101 Observer 1) or by observer name (such as Observer 2).
    c. Click OK.

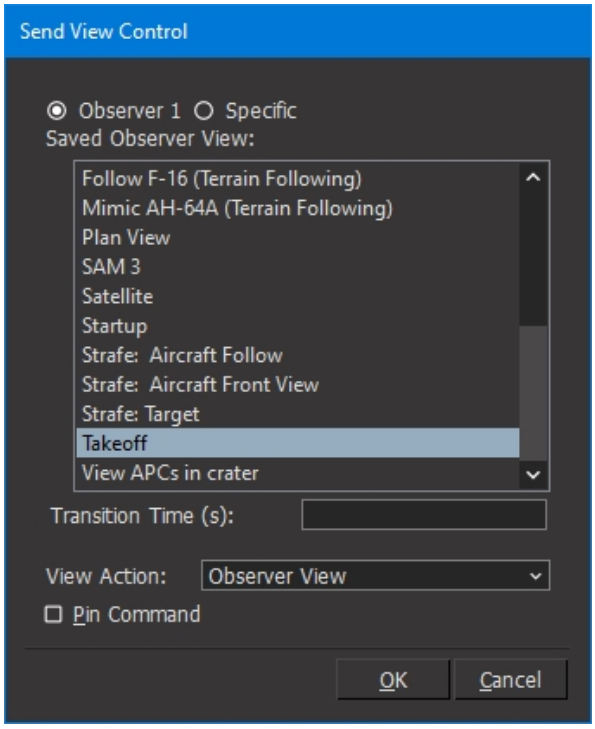
Figure 33. Observers to control



3. In the View Action list, select the view control you want to send:
   - Reset. Reset the observer view to the default view.
   - Detach. Detach the observer from an attached simulation object.
   - Observer View. Sends the selected observer view (from the Observer Views Panel).
   - *Attach mode*, where *Attach mode* is Absolute or one of the attach modes. Sends the selected attach mode.

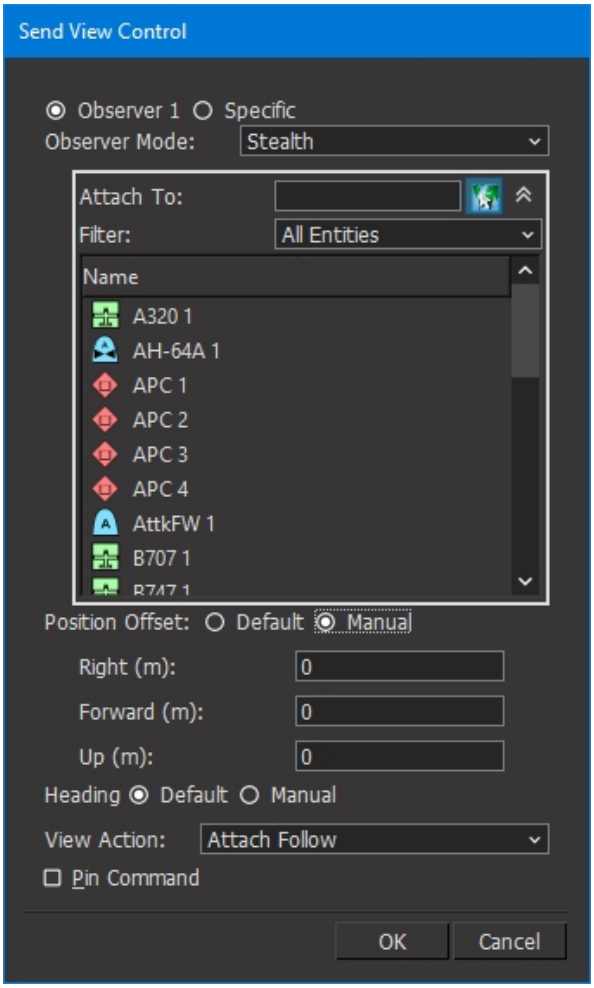
   For every command except Reset and Detach, the dialog box redisplays to add a selection window appropriate to the view mode.
4. If you selected Observer View as the View Action, select the observer view that you want to send (Figure 34). Optionally, specify a transition time, in seconds.

Figure 34. Observer view



5. If you selected Absolute mode or an attach mode as the View Action, complete the dialog box (Figure 35):
    a. In the Observer Mode list, select the observer mode for the view control message.
    b. If you selected Absolute mode, specify the location of the observer and, optionally, an offset orientation.
    c. If you selected an attach mode, select the simulation object to attach to.
    d. If you selected an attach mode that supports an offset vector or offset orientation, optionally select Manual and specify an offset. The default offset is 0,0,0.

Figure 35. Attach mode



e. If you select Track, specify the location for the observer to track from.

f. If you select Tether Track, select a secondary simulation object and, optionally, a position offset.

6. Click OK.

3.7.8 Creating a Sensor View from a Plan

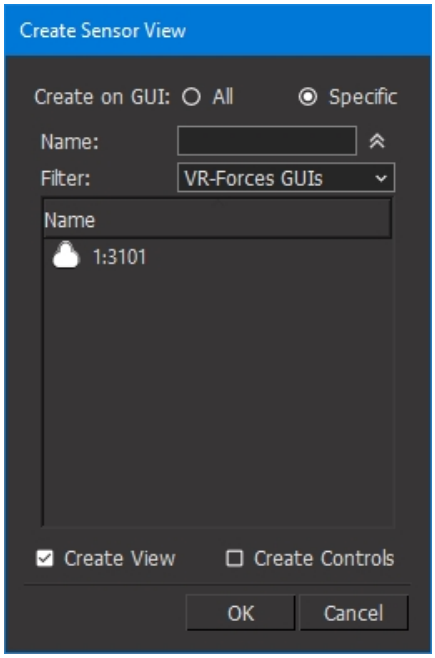
If a simulation object has a gimbaled sensor, it can create a view as part of its plan. You can specify whether the view displays in all GUIs or only a specific one. You can also indicate whether the view should include the sensor controls.

➤ **To create a sensor view from a plan:**

1. In a plan window, choose **Commands > Create Sensor View**. The Create Sensor View dialog box opens.
2. You can choose to create the sensor view in all GUIs or for a specific one. If you choose to create the view for a specific GUI, the dialog box allows you to specify which one.
3. Select the Create View check box to control whether the sensor view displays.

4. Optionally, select the Create Controls check box if you want to include view controls for the sensor view.
5. Click OK.

Figure 36. Create Sensor View



3.7.9 Sending Test Results

You can include test results in a plan for testing purposes. Test results print to the console, similar to console messages (as described in "3.7.3 Sending Console Messages" on page 44).

> You can set the sim to exit immediately after the test result is issued. For details, see the `--exitOnTestResultCommand` command-line option in Table 1.

➤ **To send a test result from a plan:**

1. In a plan window, choose **Choose > Test Result**. The Test Result dialog box opens.
2. Select the result from the list.
3. Optionally, enter a message for the test result.
4. Click OK.

3.8. Writing Plans for Units

➤ **To write a plan for a unit:**

1. Select the unit.
2. Follow the procedures for writing a plan for an individual simulation object.

> ⓘ See "Independently Tasking Unit Members" for notes about how simulation objects that are members of units respond to individual plans and independent tasks.

3.9. Writing a Plan for Multiple Simulation Objects

You can write a plan and assign it to multiple simulation objects at the same time. If any of the simulation objects already has a plan, it is overwritten by the new plan. After you assign a plan using this procedure, you can subsequently edit the plans for particular simulation objects and the changes do not affect the other simulation objects.

> ⓘ If you select several simulation objects and they all have the exact same plan, the plan window displays that plan. If any of the simulation objects have different plans, you are given the option to continue or cancel. If you continue, VR-Forces displays an empty Plan window and the new plan replaces whatever plans the simulation objects previously had.

➤ **To write a plan for multiple simulation objects:**

1. Select the simulation objects to which you want to assign the same plan.
2. Choose **Objects > Plan**. The Plan window opens.
3. Add statements to the plan as you would if you were writing a plan for an individual simulation object, as described elsewhere in this chapter.
4. Click OK. The plan is assigned to the selected simulation objects.

3.10. Writing Plans for Remote Entities

If you attach components to remote entities, you can write plans for those entities. To assign plans, the entities must exist in the exercise, must have attached components, and must be present in the exercise when VR-Forces saves the scenario. The plans must be appropriate for the attached components. Movement tasks are not appropriate, because VR-Forces cannot control the location of a remote simulation object. Appropriate plan statements might be to enable spot reporting, or to send a radio message. For details about how to attach components to remote entities, see "Attaching Components to Remote Entities".

3.11. Copying Plans and Plan Statements

You can copy plans and individual plan statements and paste them elsewhere in a plan or into another simulation object's plan. You can even paste them into another scenario if you do not close VR-Forces between the copy and paste operations.

> ⓘ In addition to copy and paste, you can drag statements and condition blocks from one location to another in a plan and to other plans.

➤ **To copy plans and plan statements:**

1. Open the Plan window for the plan that you want to copy.
2. To copy the entire plan, select the top line in the plan (**Main Plan Block**). To copy a statement, select the statement. If you select a conditional statement, the entire statement block is selected.
3. Choose **Plan > Copy Plan** or click the Copy button ( ).
4. Open the Plan window for the simulation object you want to copy the plan to.
5. Select the line just above where you want to insert the copied plan.
6. Click the Paste button ( ).
7. Continue editing the plan.
8. Save the plan.

3.12. Restarting a Plan

There may be occasions when you want to restart a simulation object's plan without restarting a scenario. You can restart a plan interactively or by placing a Restart Plan command in a plan.

When you restart a plan, VR-Forces processes the statements in the plan, starting from the first statement. However, the simulation object executes tasks starting from its current location, not the location it had at the start of the scenario (unless it has not moved since the start of the scenario).

➤ **To restart a plan interactively:**

1. Select the simulation object whose plan you want to restart.
2. Choose **Objects > Manage Plan > Restart Plan**.

➤ **To restart a plan from within a plan:**

1. Open a simulation object's plan or a global plan.
2. Choose **Commands > Restart Plan**. The Restart Plan dialog box opens.
3. Select the simulation object whose plan you want to restart.
4. Click OK.
5. Save the plan.

3.13. Abandoning a Plan

If you give a simulation object that is executing a plan an independent task, it abandons the plan. You can also order a simulation object to stop executing its plan. When you abandon a plan:

- The simulation object stops whatever task it is working on, even if it is an independent task.
- The plan is marked as complete.

- All registered triggers are deleted.
- The statement indicator skips to the end of the plan.
- The simulation object enters a wait state.

> ⓘ If you give a simulation object that is executing a plan an independent task, VR-Forces prompts you before it abandons the plan. (There is no prompt for the Abandon Plan command.) You can disable the prompt. For details, see "Enabling and Disabling the Task Confirmation Prompt".

You can abandon a plan interactively or by sending an Abandon Plan command in an individual or global plan.

➤ **To abandon a plan interactively:**

1. Select the simulation object whose plan you want to abandon.
2. Choose **Objects > Manage Plan > Abandon Plan**.

➤ **To abandon a plan from within a plan:**

1. Open a simulation object's plan or a global plan.
2. Choose **Commands > Abandon Plan**. The Abandon Plan dialog box opens.
3. Select the simulation object whose plan you want to abandon.
4. Click OK.
5. Save the plan.

3.14. Editing a Plan During a Simulation

When the simulation is not running, you can make changes to the plan. Once you save the changes to the plan, the entity returns to the first line of the plan and starts it from the beginning. This can cause its activities to be out of synchronization with the other simulation objects in the simulation.

By default, plans are locked when you run a scenario. If you want to edit a plan while the simulation is running, you must click the Unlock button (🔒). VR-Forces displays a warning message that changing the plan while the scenario is running can have unintended consequences.

4. Synchronization Matrix

This chapter describes the use of the Synchronization Matrix.

4.1. Overview of the Synchronization Matrix

The VR-Forces Synchronization Matrix (also called a sync matrix) allows you to work with multiple entities or units and add coordinating plans that are laid out in a series of phases. It is easier for you to create and manage scenarios where there is synchronization between entities or units. The intention is that you use one matrix for entities or units that are working together, rather than for all simulation objects or forces in the scenario. For example, the friendly and opposing forces would each have a separate matrix. If two groups in the same force are not coordinating their actions, they could also have separate matrices. An entity or unit can only be in one matrix in the scenario.

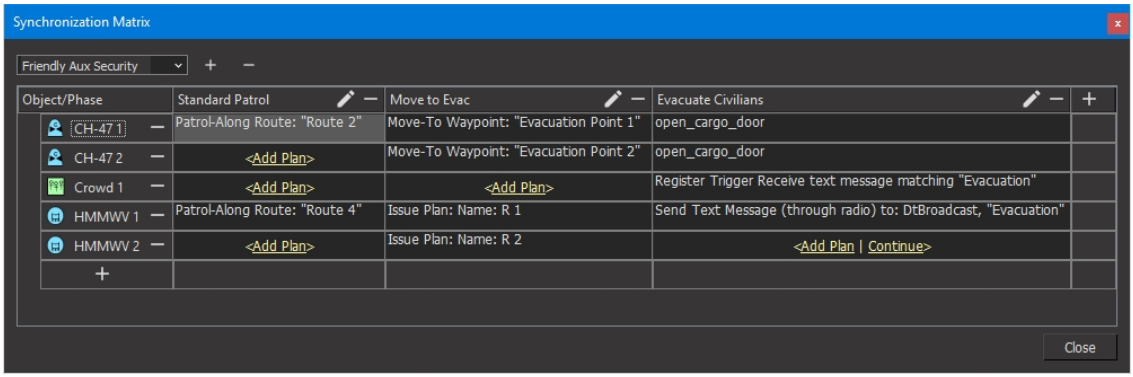
In scenarios where there is a high level of complexity, it can be hard to track down a problem where things are not happening as expected. By putting the plans for cooperating entities or units in a matrix, it is easier to see the relationships between their actions.

> ⚠ The Synchronization Matrix is another place for plans to exist. This can make it difficult to debug a scenario that includes individual plans on entities or units, a global plan, and also a collection of simulation objects with a Synchronization Matrix.

With individual plans, you can have a plan that has multiple branches based on the situation. The Synchronization Matrix, however, is intended for a linear series of actions.

You build a matrix for a particular scenario or battle plan. It is not a generic behavior that you can duplicate to another scenario or to another group of entities in the same scenario.

Figure 37. Synchronization Matrix Example



4.1.1 How the Synchronization Matrix Works

A VR-Forces Synchronization Matrix is a simulation object that uses the Issue Plan command to send the plans in the matrix to the entities or units. As a phase progresses, each entity and unit in the matrix receives its plan. The plan can be a simple task or it can be more complex. For example, you can set ordered speed and start an entity moving in the same phase. You could also have a reaction where an entity or unit detects an entity, moves to attack, and returns.

Each phase has a Phase End Condition, and when that condition is reached then the phases ends and the next phase begins. When a phase reaches its completion, all plan information from that phase is discarded and the plans in the next phase are used. See "4.2.3 Adding and Editing Phases" on page 59 for instructions.

4.1.2 The Synchronization Matrix Is an Object

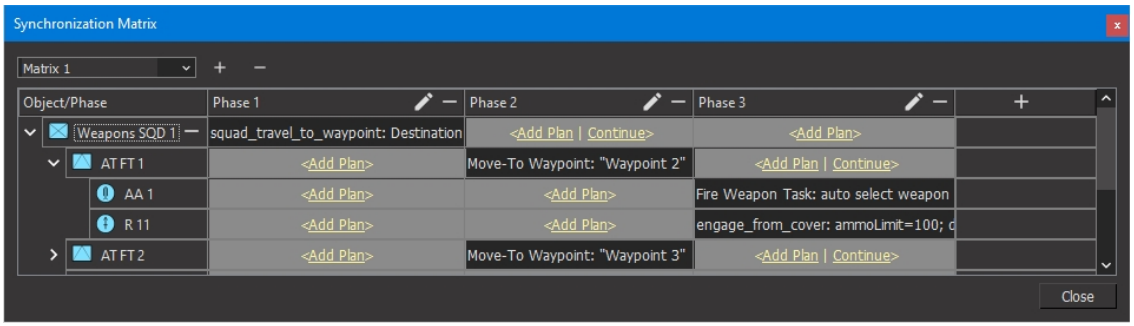
The Synchronization Matrix is a VR-Forces object and it exists on a single sim engine. A synchronization matrix can be on a different sim engine than other simulation objects. If the scenario has multiple matrices, you could put each matrix on a separate sim engine.

The Synchronization Matrix sends the issue plan commands, runs conditional tests, and monitors for phase completion.

4.1.3 Units and the Synchronization Matrix

A synchronization matrix can contain both entities and units. (It can also include part of a unit if that is appropriate for your scenario.) You can include a multi-level unit in a matrix. In the matrix, you can expand a unit to see the sub-units or entities. In each phase, you can make a plan for a unit or the entities that comprise the unit. You cannot do both in the same phase.

Figure 38. Synchronization Matrix for Multi-Level Unit



4.1.4 Spot Report Viewpoint in the Synchronization Matrix

When you configure VR-Forces to show the ground truth for only some of the forces, objects in the other forces are hidden from the Synchronization Matrix unless they are spotted.

These figures show the Synchronization Matrix when Ground Truth for Opposing force is disabled. In Figure 39, the Friendly force has not detected any members of the Opposing force. In Figure 40, the Friendly force has spotted (and destroyed) some members of the Opposing force.

Figure 39. No contacts

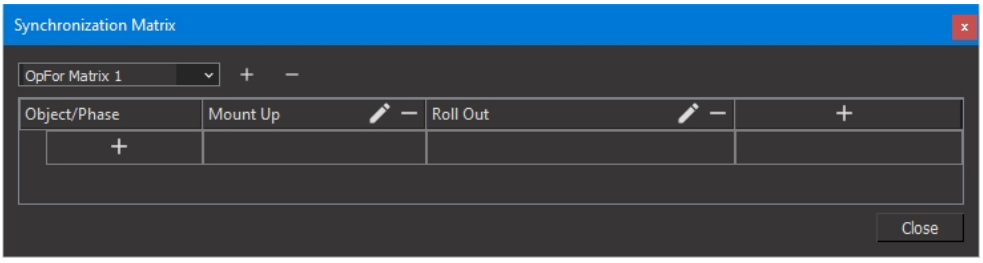
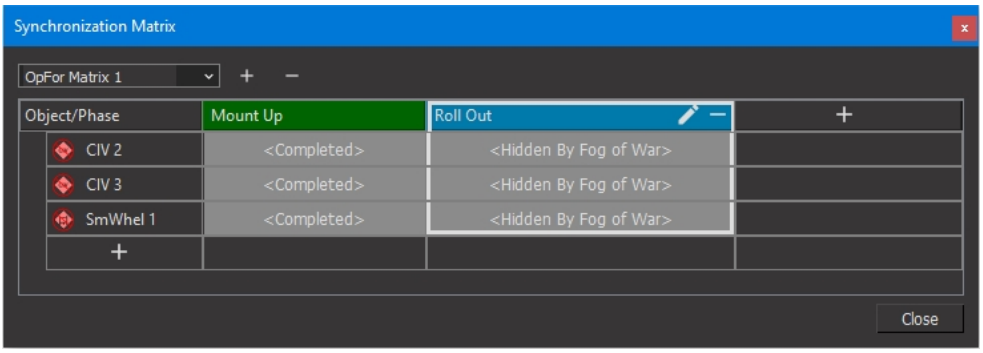


Figure 40. Some contacts



See "Configuring the Spot Reports Viewpoint" for details on spot reports and fog of war.

4.2. Building a Synchronization Matrix

Building a Synchronization Matrix involves:

1. Creating a matrix.
2. Adding entities and/or units to the matrix.
3. Adding phases to the matrix.
4. Adding plans for the entities and units.

4.2.1 Opening and Creating a Sync Matrix

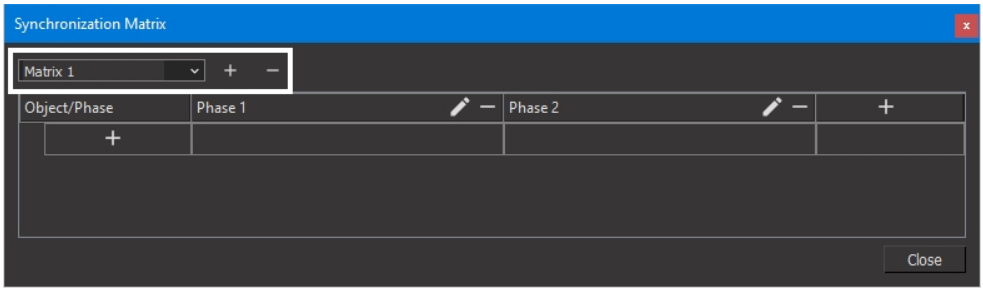
The Synchronization Matrix dialog box (Figure 41) allows you to manage each sync matrix in your scenario. You use it to add entities or units to the matrix, manage their plans, and add and edit the phases in the matrix.

When you open the Synchronization Matrix for the first time in a scenario, it is blank. If you have one or more matrices in the scenario, the first matrix displays.

➤ **To open the matrix, do one of the following:**

- Choose **Simulation > Synchronization Matrix**.
- Click the Open Synchronization Matrix button (#) in the Scenario Components toolbar or the Last Selected Object Panel.

Figure 41. New Synchronization Matrix



➤ To create a new matrix, click the Add Sync Matrix button (+).

➤ To view or edit a matrix, select it from the list.

➤ To change the name of the matrix, simply type in the text box.

➤ To delete the currently selected matrix, click the Remove Sync Matrix button (−).

4.2.2 Adding Entities and Units

To create the Synchronization Matrix, you need to add the entities or units that will be working together. The matrix can include entities and units.

➤ **To add an entity or unit to the matrix:**

1. In the Object column, click the Add Object button (+). The cursor turns to a hand.
2. Select the entity or unit in the Objects List or on the terrain. It is added to the matrix.

You can also remove an entity or unit from the matrix.

➤ To remove the object, click the Remove Object button (−) next to the entity or unit's name.

4.2.3 Adding and Editing Phases

A new Synchronization Matrix starts with two phases. You can add more phases as needed.

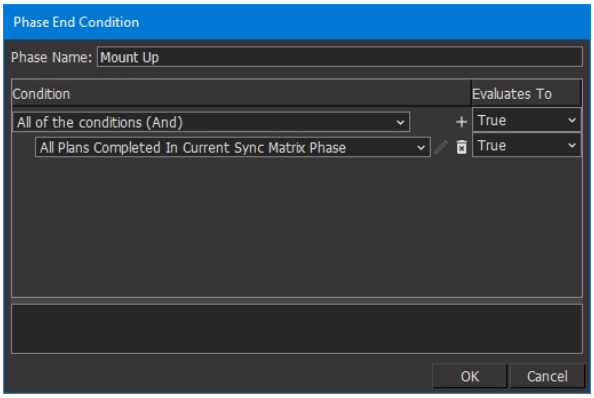
➤ To add a phase, click the Add Phase button (+) at the top of the right-most column of the matrix.

You can edit a phase to specify the phase end condition. The default end condition is All Plans Completed in Current Sync Matrix Phase. In that case, the next phase begins only when all entities and units in the matrix have completed their plans in that phase. You can also specify another condition. See "2.2. Conditional Statements" on page 5 to understand more about conditions. When the phase end condition is met, the Synchronization Matrix moves to the next phase.

➤ **To edit a phase:**

1. In the header for the phase, click the Edit button (✎). The Phase End Condition dialog box opens (Figure 42).

Figure 42. Phase End Condition



2. Change the condition to meet your needs. See "3.4. Building Condition Statements" on page 28 for more information.
3. If desired, change the Phase Name.
4. Click OK. You return to the Synchronization Matrix.

You can also remove a phase from the matrix.

➤ To remove the object, click the Remove Phase button (—) next to the phase name and then click Yes in the confirmation message.

4.2.4 Adding and Editing Plans

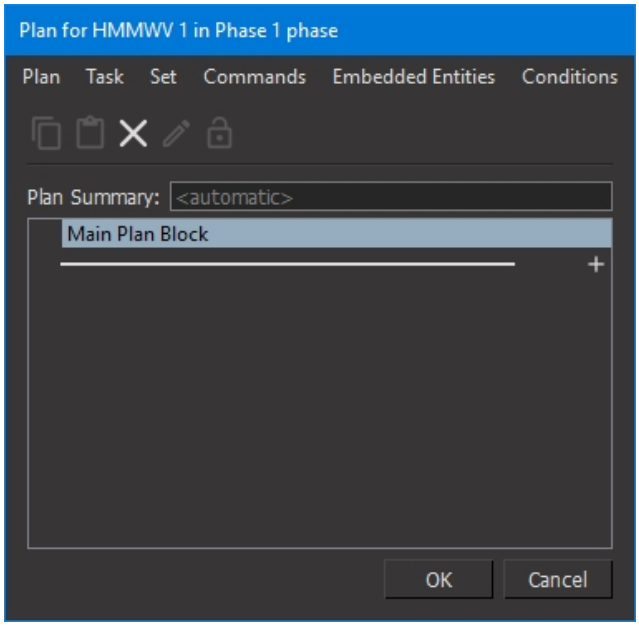
Each cell in the Synchronization Matrix can hold a plan for the associated entity or unit.

➤ **To add or edit a plan, do one of the following:**

- Single-click in a cell that reads Add Plan.
- Right-click in a cell and choose **Set Plan** or **Edit Plan**.

The Plan for Phase dialog box opens (Figure 43).

Figure 43. Plan for Phase dialog box



Use the Plan for Phase dialog box to create the entity or unit's plan. See "Chapter 3. Writing Plans" on page 19 for more details.

The Plan for Phases includes a Plan Summary box where you can enter a description of the plan to display in the Sync Matrix.

4.2.5 Copying and Pasting Plans

You can copy a plan from one cell in the Synchronization Matrix and paste it in another.

➤ **To copy and paste a plan:**

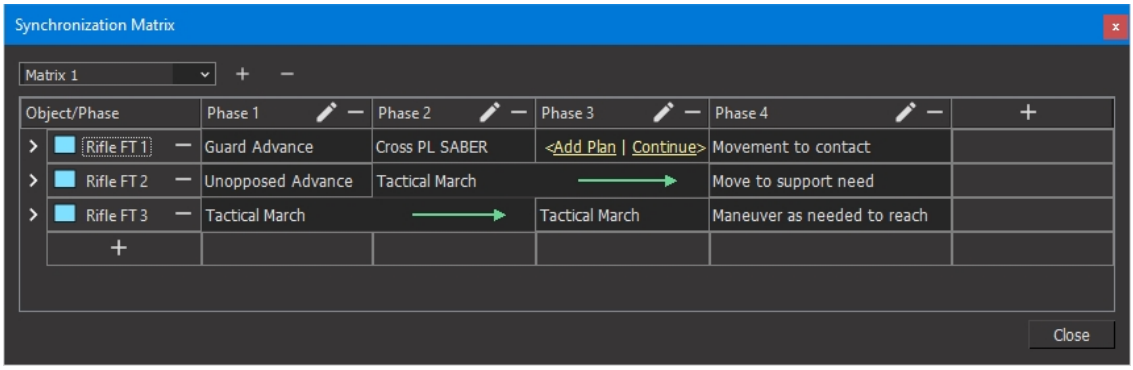
1. Right-click on a cell with a plan and then choose **Copy Plan**.
2. Right-click on another cell and the choose **Paste Plan**.

4.2.6 Continuing a Plan

Sometimes, you might want an entity or unit's plan to continue across two or more phases. This allows one phase to end and the next phase to begin while the entity or unit continues to perform its plan, rather than abandoning if there is no plan set. A phase with a plan marked to continue is considered complete for the phase end condition test.

➤ To continue a plan through a phase, click Continue in the cell for that phase. An arrow displays, showing that the plan spans that phase as well as the phase where it begins.

Figure 44. Synchronization Matrix with continued plans



You can also discontinue a plan that was previously set to continue through a phase.

➤ To remove the continuation of a plan, right-click on the arrow and then choose **Discontinue Previous Plan**.

4.2.7 Deactivating and Activating Plans

You can deactivate plans for units. When a unit's plan is deactivated, the subordinates perform their own plans.

➤ To deactivate a plan, right-click on the plan for a unit and then choose **Deactivate Plan**.

You can also reactivate a plan that you had previously deactivated.

➤ To reactivate a plan, right-click on the unit's plan and then choose **Reactivate Plan**.

4.2.8 Removing Plans

➤ To remove a plan, right-click on a plan and then choose **Remove Plan**.