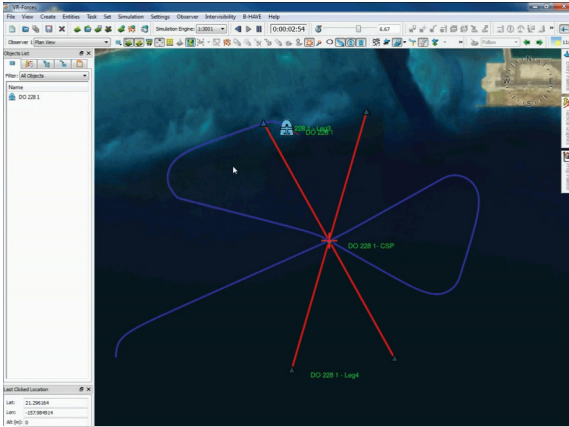


What's Up MÄK

Scriptable Tasks - Is Faster Really Better? Yes!



Just because something is faster doesn't necessarily make it better. We're here to prove that faster really is better when it comes to creating complex behaviors for a CGF. Scriptable tasks in VR-Forces give you the power to quickly develop complex tasks, easily coordinate group behaviors, and script GUI components in minutes, empowering you to develop better and more compelling simulations.

Rapid development cycles let you take advantage of the often limited time you have with subject matter experts (SMEs). Together you can transcribe the problem into behavior, test it interactively, fine tune it, and make it right. The more reliable information about the problem that you can encode into the behaviors, the more valid your simulation will be – more iterations means a higher quality result.

Read how MÄK Engineers are getting creative with **scriptable tasks**

Lets try an example

Let's say you've been tasked to develop a search and rescue mission – you have limited time and know little about the actual search and rescue patterns or protocol.

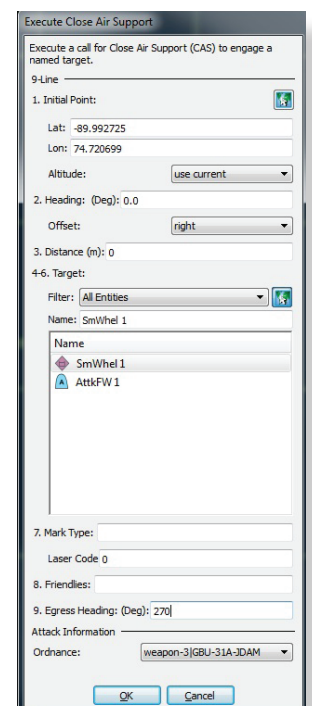
You decide to consult the WSDOT Aircrew Training Text as your expert. After some research, you learn about the different visual search patterns and you throw together a quick script that incorporates a specific pattern, and then you test it out. 20 iterations later, paired with a little feedback from SMEs, and you probably have a pretty good script consisting of several search patterns embedded in one another. Maybe it even looks like the graphic at the top of this page.

Jim Kogler, MÄK's Director of COTS Products, created this same scenario from the ground up. Interested in seeing the end result? Check out a [quick video](#) showing a sector search scenario made with Lua scripting. (For examples of actual Lua scripting code, check out our [community forum](#) for scenarios our engineers created.)

Interfaces that match the mission

Describing behavior is one thing, but then there's the challenge of building and integrating the user interface into the CGF. VR-Forces makes this easy. Scriptable tasks auto-generate the GUI and attach it to the appropriate entity types in the Task menu.

Lets say you want to make a Close Air Support task that uses a standard 9-Line briefing for the interface. It's now possible to quickly create and edit a GUI to match your mission, no programming required. Using scriptable tasks, MÄK has developed a sample 9-line to show you how its done (right). You can even edit it on the fly, while the simulation is running, if need be.



MÄK is here to help

Complicated processes that used to take weeks – even months – with C++ coding are reduced to hours, thanks to scriptable tasks. The simplicity of scripting tasks in VR-Forces enables users with basic programming skills to tackle the most daunting simulation challenges. Let MÄK help you get the most from your CGF.

Scriptable Tasks in VR-Forces : 1

MÄK's Tech Tip by Fred – Ending VR-Forces Scripted Tasks : 2

MÄKer Spotlight: Alan Kosow : 2

RT Dynamics' FixedWingLib and RotorLib Plug-ins for VR-Forces : 3

MÄK's Tech Tip by Fred Wersan

Ending VR-Forces Scripted Tasks

The big news in VR-Forces 4.1 is the scriptable task feature. It lets you add new tasks written in the Lua programming language. This is the first of several tech tips that will cover various aspects of how to write tasks for VR-Forces.

When you write a Lua script for a scripted task, you end it with the `vrf:endTask(true)` function. However, if you don't understand how it works, you might be surprised at the results. Consider the following simple pseudo-code:

```
somecondition = false
function tick()
    if somecondition == true then
        vrf:endTask(true)
    end
    -- do some stuff
    somecondition = true
end
```

You would expect that the first time through `tick()`, `somecondition` is false and the script would do some stuff. The second time through, `somecondition` is true and the task would exit before it did anything else. However the actual output of the script would look like this:

```
Starting task.
Doing some stuff.
Task Complete.
Doing some stuff.
```

Why did it do more stuff after it ended? The answer is in the definition of `endTask()`: "Causes the current task to end once the current tick is complete."

As Yogi Berra said, "It ain't over 'til it's over." The `tick()` wasn't complete until it did some stuff again, which is why it got repeated. To avoid this, use the following code:

```
if somecondition = true then
    vrf:endTask(true)
else
    -- do some stuff
    somecondition = true
end
```

This isn't the only approach to ending a scripted task. For other examples, take a look at the scripted tasks we provide with VR-Forces at the [Lua Script Exchange](#) in the MÄK Community Forum. As Shakespeare said, "All's well that ends well."

Free VR-Forces and VR-Link Training Course with Purchase of Maintenance!

Check out the [MÄK Events Calendar](#) to learn about our next training course! Have questions? Contact your sales person or info@mak.com.

MÄKer Spotlight: Alan Kosow

Account Manager

Many of you already know Mr. Alan Kosow, one of MÄK's great account managers. Alan is known for his skill in building relationships with customers, uncovering new opportunities and then coordinating the successful completion of evaluations, negotiating purchase agreements, and going above and beyond to ensure customer satisfaction.

Now approaching his fourth year at MÄK, Alan came to us with 20 years of experience in the software business, 10 years' experience as an IT director, and a Bachelor's degree in electrical engineering. His technical expertise makes him a huge asset to our MÄK team!

While Alan is obviously skilled in the workplace and dealing with customers, there are several things you might not have known about him. For example, Alan has a green thumb – he's both an indoor and outdoor gardener, tending mostly to his orchid collection (in addition to mending coworkers' ailing office plants).

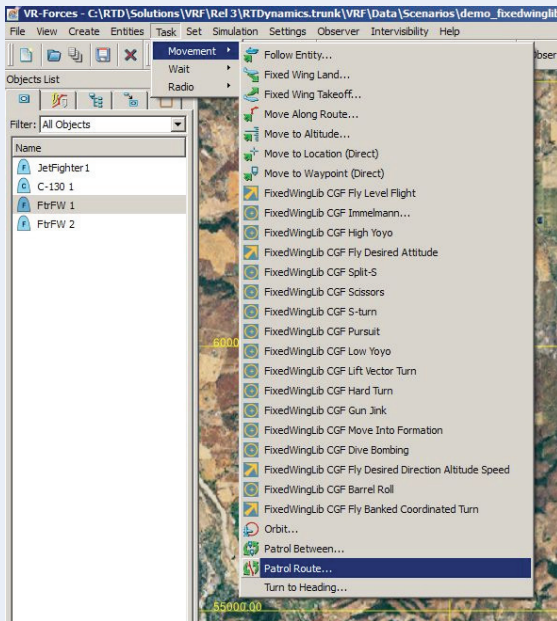
Another thing you may not have known is that Alan is an active part of the Ham Radio community – also known as Amateur Radio. (Basically, Ham Radio is a popular hobby in which licensed participants operate communications equipment, with a deep appreciation of the radio art.)

We hope we've piqued your interest in getting to know our MÄK team. Stay tuned to our [website](#) for the latest on MÄK!



Support for High Fidelity Fixed and Rotary Wing Aircraft for VR-Forces Now Available!

RTDynamics, MÄK's partner company, has released FixedWingLib and RotorLib for VR-Forces 4.1



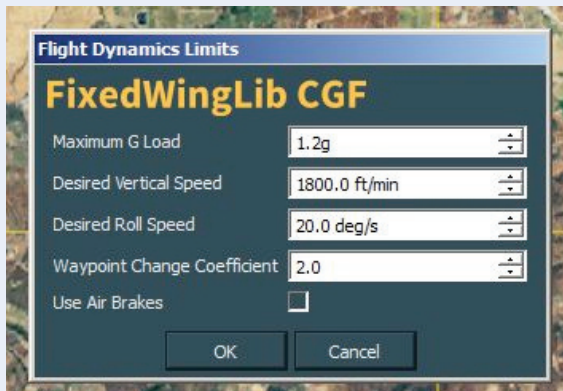
RTDynamics has developed a ready-to-use VR-Forces plug-in that integrates FixedWingLib and RotorLib into VR-Forces, giving developers high fidelity aircraft models with 6 degrees of freedom (DOF) for realistic aircraft maneuvers. The plug-in introduces new available Task and Set data request types for FixedWingLib and RotorLib CGF aircraft entities.

RTDynamics' high fidelity air model plug-ins enable users to build realistic scenarios, such as air combat and close air support, in VR-Forces with no programming required. The plug-in provides new tasks out-of-the-box such as "Fly Desired Attitude", "Fly Coordinated Turn", "Level Flight", "Fly Desired Altitude-Speed-Heading", and "Terrain Following", which can be combined in VR-Forces Plans to simulate any airborne mission – at a higher level of fidelity than previously possible. Users can also choose to create their own high fidelity aircraft maneuvers with RTDynamics aircraft models using scriptable tasks with Lua, now available in VR-Forces 4.1. (To learn more about scriptable tasks, check out our [feature story](#) or contact info@mak.com.)

Aircraft in VR-Forces can be automated through plans or controlled interactively with a joystick, allowing the user to control exactly how a virtual pilot executes a desired maneuver. To determine the agility of a break turn maneuver, for example, users simply need to set the maximum G load and performance parameters.

The FixedWingLib and RotorLib plug-ins come with an optional Combat Maneuvers Library add-on. This package of ready-to-use combat maneuver tasks gives VR-Forces the ability to perform vertical take-off and landing (VTOL) maneuvers, formation flight, and terrain following for fixed-wing and rotor-wing aircraft.

RTDynamics' FixedWingLib and RotorLib for VR-Forces allow developers to take advantage of the simplicity of a plug-in to build complex aircraft maneuvers. If more flexibility and modifications are required, both plug-ins come with a software development kit for user customization.



For additional information please contact devrim.erdem@rtdynamics.com or info@mak.com.

