

VsTASKER- A Simulation SDK Tool to Build simulators | AEROPVS01

Leverage a unique white box SDK to build your next simulation system. By combining visual paradigm and code generation, it is even easier than before to design new models, define bespoke behaviors, assign plans to specific entities and create all kind of scenarios, from a simple test-bed to the most complex distributed trainer. Regular Updates For more than 15 years, vsTASKER has evolved to adapt to the changing requirements of many customers around the world.

vsTASKER is a professional, real-time simulation development toolkit developed by **VirtualSim**. It functions as a "white box" synthetic environment framework, allowing engineers and developers to design, build, and deploy complex logic and models. The software is heavily utilized across defense, aerospace, and commercial industries for tactical military simulation, autonomous systems testing, and traffic engineering.

Core Architecture

The toolkit is structurally split into two primary environments:

- **The GUI (Design Environment):** A graphical visual programming platform where users build scenarios, map multi-layered terrain data (GIS), design state machines, and drops specific object-oriented entities onto a 2D or 3D canvas.
- **The SIM (Simulation Engine):** An independent simulation engine compiled by Microsoft Visual Studio. It is built directly from the C++ code automatically generated by the GUI. The SIM engine runtime is license-free, making standalone deployment highly scalable.

Key Features

- **Automatic Code Generation:** Translates visual paradigm diagrams and logic into high-performance C++ code automatically, eliminating manual file-handling bottlenecks.

- **Space & Orbit Propagators:** Features built-in Keplerian, SGP4, and J2 propagators to simulate satellites and space vehicles using data directly from NORAD Two-Line Element (TLE) descriptions.
- **Interoperability:** Connects natively with industry-standard third-party game and graphics engines (like Titan Vanguard or VBS-IG) using CIGI or HLA network simulation architectures.
- **Real-time Data Streams:** Features modules capable of importing real-time live data streams, such as tracking live aircraft flights via ADS-B streams.

Common Use Cases

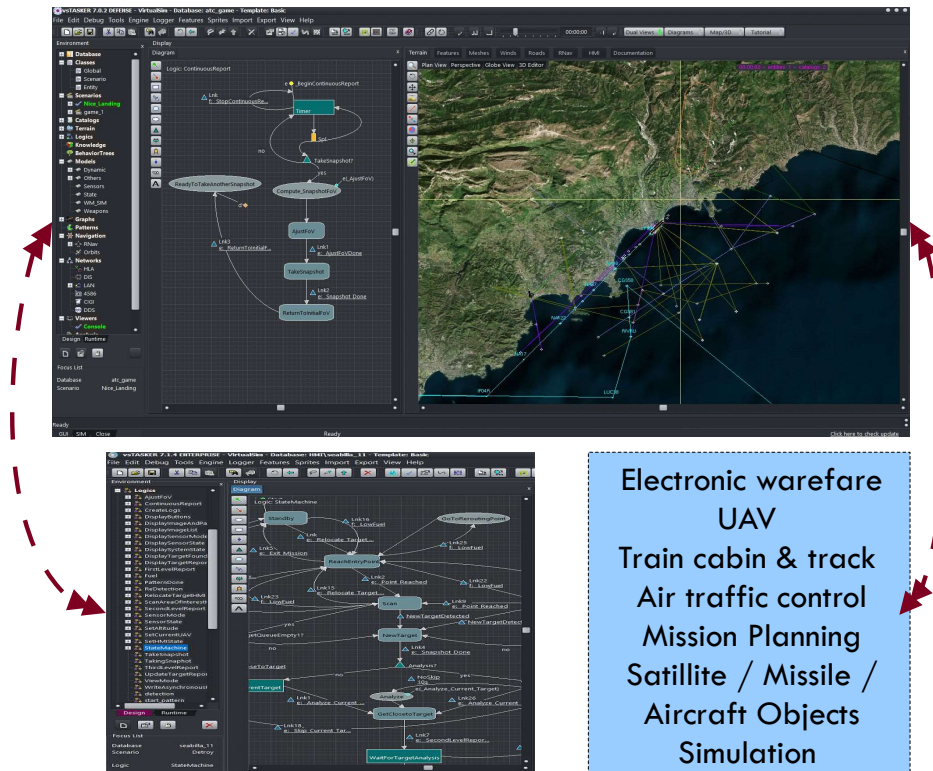
- **Military & Defense:** Unmanned Air System (UAS/UAV) autonomous behavior, mission planning, radar sensor physics verification, and tactical team trainers.
- **Aviation & Space:** Air Traffic Control (ATC) terminal radar approach simulation, multi-station pseudo-pilot coordination, and satellite orbit shadowing computations.
- **Transportation Systems:** Metro line train cab mockups, automatic heavy-truck trailer parking physics, and commercial maritime port traffic analytics.

If you are looking to deploy or build a project with this toolkit, tell me:

- What **type of scenario** you are trying to simulate (e.g., UAV automation, orbital paths, or land traffic)?
- Which **graphics engine or API** you plan to connect it to?
- Whether you need help setting up the necessary **Visual Studio compiler paths** or environment variables.

Free simulation engine

Design your system from scratch or templates, generate the code and use it free of charge. Buy once and use it for all your work



vsTASKER allows study, animation, and visualization of any size and complexity of scenarios. Using graphical paradigm and automatic C++ code generation, user can build a simulator with unmatched simplicity. vsTASKER does not limit to any particular field (see some projects) Targeting software engineers and system integrators, it provides under one roof everything needed to develop a simulation infrastructure, maintainable and scalable. It covers a wide range of needs, from Defense, mission planning, CGF, electronic warfare, UAV, train cabin simulators, railways and ATC trainers (...) After 15 years on the market worldwide, vsTASKER has proven its value under high technical constraints and tight schedules

vsTASKER is a development environment for real-time simulation. It allows design, study, animation and visualization of complex logic and models with unmatched simplicity. Based on a modular architecture, it tremendously helps simulation engineers to address from simple studies up to high-end systems.

VsTASKER Features

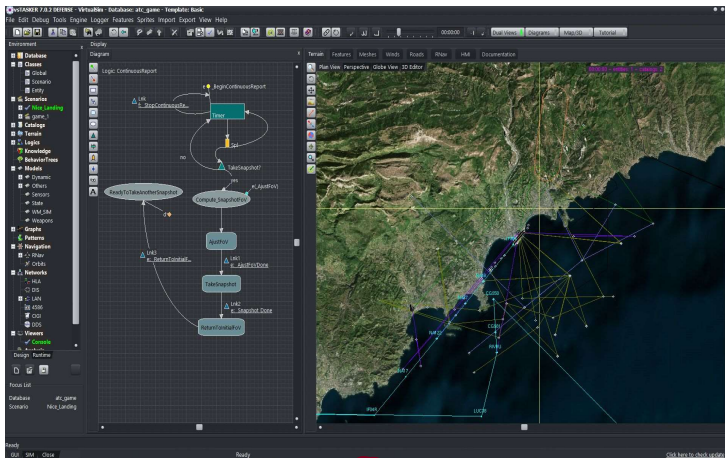


Diagram based

Draw your logic using sequential function charts and let the tool translates all graphical diagrams into high performance C++ code. Specific runtime mechanisms are automatically code generated and reduce drastically development time and errors.

Engineer friendly

You can directly pinpoint your code inside the simulation, removing the burden of working with numerous files and folders. You press a button and your code is part of the system. Moreover, you can follow all behaviors during runtime and interact with all components.

Free simulation engine

vsTASKER generates a free simulation engine ready for deployment. You do not need to pay for runtime license. Your team can address several projects with the same SDK for the same price.

Code generation tool

As a workbench, vsTASKER allows you to gather user-code, libraries, components, DLL, third-party software, Matlab functions complex logic and all kind of materials into a unique and intuitive graphical user interface. *in this figure, we see the development cycle around vsTASKER, from the design phase in the GUI editor down to the Simulation Engine in vsVIEWER. Once defined, the cycle is automatically repeated on a mouse click.*

White Box

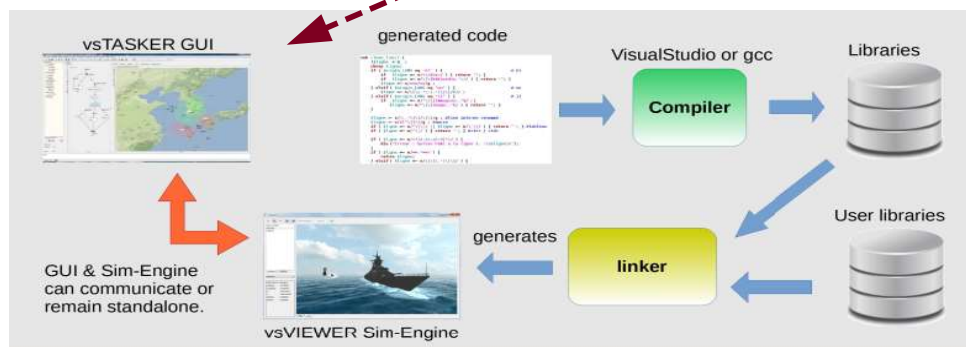
With a documented API and source code of all components and DLL, you can expand or customize the product to exactly fit your needs

Batch mechanism

The RTC has been designed for maximum performances. 30Hz simulation can be accelerated to maximum speed without loss of accuracy. You can run thousands of times a scenario and change values according to your laws to analyze results with built-in charts.

User Parameters

When building your logic and models, you can specify which variables are dynamics and vsTASKER will automatically generate the user-interface and update mechanisms. You can also define graphical curves for accurate input responses.



vsTASKER is a high-performance, real-time simulation development framework used primarily by defense contractors, aerospace engineers, and training system developers to build complex, synthetic virtual environments. Unlike static simulators, it functions as a comprehensive software development kit (SDK) that automatically generates optimized C++ code from graphical models

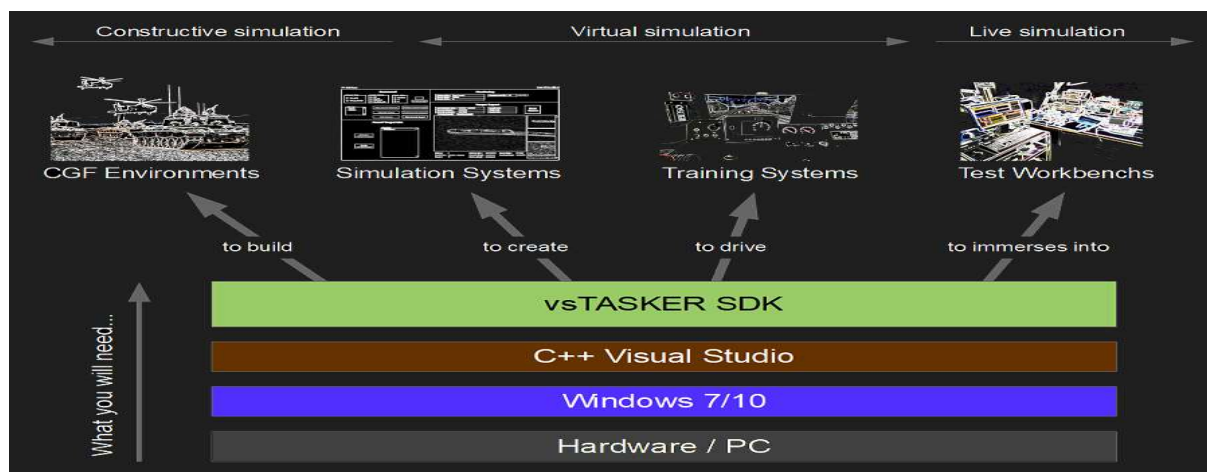
Perfect editor for complex scenarios

You need to design realistic scenarios with many agents and AI behaviors ? vsTASKER is here to help. You can first test your algorithms on the 2D map, accelerate the time, output all data you need, then validate the whole in 3D, either embedded into OSG or using DIS/CIGI protocols with your favorite IG.

Perfect Middle-Ware Solution for Integrators

vsTASKER provides unprecedented mechanisms to tailor your synthetic environment, either from a simple 2D plan map to the most complex 3D game like environment. Once you have understood the concepts, testing a scenario or developing a system becomes so straight forward you will wonder how you did before.

vsTASKER has been designed after years of consultation with customers struggling with either too low-level compilers or specialized high-end overkill products. Instead of reinventing the wheel by writing a framework from scratch or trying to break into pieces a complex expensive software to fit it into your needs, vsTASKER provides all the necessary mechanisms for real-time simulation without overwhelming you with tons of specific capabilities and parameters.



Unleash the power of a framework

While a end-of-line product only gives you what it has, a framework helps you to get what you want. See how vsTASKER main features can help you doing things by yourself:

-  Logic defined visually using Grafcets and State-Charts;
-  Knowledge based on Facts and user Rules;
-  Behavior-Trees to trigger Logic and Knowledge;
-  Component/Device models to provide capabilities;
-  Routines as piece of code dropped on the map;
-  Visual feedback at runtime;
-  Runtime mission planning;
-  Predefined material ready to be used;
-  OpenScenGraph & osgEarth ready;
-  Pin-point your code, drag & drop objects, generate code with a mouse-click

vsTASKER is the perfect solution for any customer who needs a versatile tool for his simulations. Whether it is for Defense or Civilian projects, the open infrastructure of the product will satisfy all needs on time and within budget. Many integrators have trusted us over the years with difficult cases and have managed to deliver successful results.

VISUAL PARADIGM

Design your logic graphically using the visual editor and let the code generator do all the coding for you, efficiently and without errors. State machine and behavior trees will speed up your scenario creation with unmatched simplicity. Design, assign, drag & drop, test and deploy.

PLUG & PLAY DISTRIBUTION

Quickly distribute your simulation between various players or use third party graphic engines such as VBS-IG for top quality 3D output. vsTASKER provides several different means of communications: HLA, CIGI, DDS, DIS, LAN

REAL-TIME WHITE BOX SDK

Use the published and documented API to control all aspects of your simulation. Easily add your own models with parameters or extend the built-in ones very simply. Define the entity capabilities by attaching as many models as needed using point & click.

HIGH PERFORMANCE ENGINE

Our simulation engine can handle complex logic in real time from 1hz to high frequencies (> 1000hz) and thousands of different entities can run simultaneously inside a scenario.

PLUG & PLAY DISTRIBUTION

Quickly distribute your simulation between various players or use third party graphic engines such as VBS-IG for top quality 3D output. vsTASKER provides several different means of communications: HLA, CIGI, DDS, DIS, LAN

FRONT END UI BUILDER

Drag & drop predefined objects to easily design your own GUI to interact with your simulation, without having to use external software or integrate with Qt or wxWidgets. Quickly build your OpenGL realistic user interfaces and deploy license free.

SEAMLESS SPIRAL DEVELOPMENT

Design your system database, create your behaviors visually, setup your scenarios using maps, drop entities into your terrains and give them plans, then generate the license free simulation engine.

Enjoy the Creation with tremendous results

1 Maximum versatility for customers

Addresses all kind of simulations but not only that! With its unique code generator based on simplified diagrams, vsTASKER has proven its value in CGF Environments, Simulation & Training Systems, Test Workbenches and Control Panel design.

2 Pay once & use many times

Since vsTASKER is a development toolkit, once you own the software, you can address many projects; the generated engine is license free and can be deployed without limitation.

3 Proven technology

For over 15 years in several countries, by big corporations down to small integrators, vsTASKER has proved to be the tool of choice. Engineers will enjoy having a product which provides abstraction levels and let them free to code and not only tune parameters.

Develop some AI

Open the logic editor, sketch your state-machine, write some functions or use the predefined ones, connect inputs and outputs to the environment, compile, run and see the result. Modify input values and see the result in the behavior.

Matlab/Simulink models Interface

Just load your files/models into vsTASKER, connect the I/O and see it running in real-time.

GIS/sand-table for your CGF or C3I

Use OSG or any C++ compatible third party visual engine and drive it from inside vsTASKER runtime. Your IG is compatible with CIGI protocol? Use vsTASKER direct mechanism to connect to it.

Develop some AI

Open the logic editor, sketch your state-machine, write some functions or use the predefined ones, connect inputs and outputs to the environment, compile, run and see the result. Modify input values and see the result in the behavior.

Test a military doctrine

Open a map, drop some entities on it. Give each of them some capabilities by attaching specific devices/components like a radar, a main gun for a tank, some missiles for a plane fighter, some gun for infantry (...) Then provide some dynamics, some shields, some radios and make them act on the scenario using logic, do-list or scripted plans.

Hardware in Loop Testing

Develop into vsTASKER all the drivers to the third-party elements, using DLL or LAN or HLA. Write some visual logics for initializing, synchronizing and actuating all systems and ask vsTASKER to become the brain master chief of your system.

Display your simulation in 3D with graphics

Open the logic editor, sketch your state-machine, write some functions or use the predefined ones, connect inputs and outputs to the environment, compile, run and see the result. Modify input values and see the result in the behavior.

FREE VERSION	PLAYER
Full version, without a need for an evaluation license, but with hard coded limitation on every features. Perfect to test the product before buying. The free version is capable of running many provided read-only demos	Full version at a fraction of the price but only capable of running developed scenarios. Used mainly for deployment when only the GUI is needed to control the simulation.
ACADEMIC	ENTERPRISE
This version is only provided to some universities with which VirtualSim has signed an agreement. The version is not limited but cannot be extended with modules. It is good enough for most academic purposes.	Core licensed version which covers most of the users needs in civilian and defense simulation. Optional modules can be bought separately to add more features to the software. In some countries, modules cannot be bought separately and are embedded directly into the Enterprise version. See your local distributor for more details.

OPERATING SYSTEM SUPPORT	INDUSTRY APPLICATION
HARDWARE SUPPORT	ORDERING DETAILS

Note: ** Specifications are copied from manufacturer data. Please contact before buying.