

Titan Vanguard Integration : AEROPVS014

The vsTASKER Titan Vanguard Integration embeds the vsTASKER simulation logic engine directly inside Calytrix/Titan.IM's Titan Vanguard game engine loop using a native C++ API DLL host connection. This native integration allows vsTASKER to serve as the runtime scenario creator, computer-generated forces (CGF) artificial intelligence, and tactical brain, while Titan Vanguard acts purely as a high-fidelity visual and physics game engine.

Key Architecture & Mechanics

- **DLL Injection and Direct C++ Control:** Unlike network-heavy client/server protocols (such as DIS/HLA or CIGI) used in other integrations, vsTASKER runs natively within Titan Vanguard via the C++ API.
- **Separation of Concerns:** vsTASKER handles all high-level logic, user waypoint rules, complex entity behaviors, and script orchestration. Titan Vanguard parses these commands to visually animate graphics, terrains, and localized physics.
- **Synchronization via vsTASKER 7:** Modern versions utilize synchronous code generation natively built for IDEs like Visual Studio. Developers can model entity types or behaviors inside the vsTASKER graphical interface and write complex logic structures directly inside Visual Studio, keeping both engines fully aligned.

Demonstrated Capabilities

Official VirtualSim implementations demonstrate massive scenario scaling capabilities, including:

- **High Entity Counts:** Running complex multi-domain environments featuring hundreds of individual moving soldiers, active minefields, and surface gunners.
- **Hybrid Control States:** Synchronizing automated AI logic alongside real-time human inputs, such as manual joystick-controlled helicopter flight models.
- **Integrated Exercises:** Acting as the underlying scenario editor where complex operational directives (like indirect frigate fire or patrol routes) are executed instantly within the virtual globe sandbox.

Would you like to explore the step-by-step tutorial manual for setting up the environment, or are you looking for details on how to compile code and link the custom C++ DLL between the platforms?

The **vsTASKER Titan Vanguard Integration** allows developers to embed the vsTASKER simulation and AI behavior engine directly into the Titan Vanguard 3D world visual platform. Rather than connecting them across a slow network via client/server protocols, this integration embeds the AI logic straight into Titan's core loop using a dynamic-link library (DLL).

This setup handles heavy battlefield simulations by treating Titan Vanguard as a high-fidelity visual engine, while vsTASKER directs all the underlying tactics, logic, and entity movements in real time.

How the Architecture Works

- **Plug-and-Play DLL:** vsTASKER generates a custom compiled C++ DLL that Titan Vanguard loads automatically at startup.
- **Direct API Control:** The runtime simulation DLL takes direct command of Titan Vanguard's native C++ API.
- **Real-Time Injection:** It programmatically spawns entities, sets coordinates, and updates object behavior frame-by-frame inside the 3D scene.
- **No Network Lag:** Eliminating external communication protocols (like HLA/DIS or network sockets) maximizes performance and processing speeds.

Key Technical Capabilities

- **Massive Scenario Scale:** Capable of flawlessly driving massive scenarios—such as hundreds of moving soldiers, hidden minefields, automated gunners, and user-controlled air assets simultaneously.
- **Mixed Autonomy:** Allows scripting complex waypoint paths for AI units while leaving other assets free to be steered manually by pilots using joysticks or gamepads.
- **Visual Scenario Editing:** Developers can utilize vsTASKER as an offline scenario builder to model behavioral trees, which then inject effortlessly into Titan's global sandbox.

Common Applications

- **Tactical Military Training:** Creating realistic "what-if" combat scenarios, combined arms drills, and automated enemy behaviors.
- **Joint Fires & JTAC Simulator Support:** Injecting smart AI entities into large-scale projection systems and dome setups managed by Calytrix Titan Vanguard software.
- **Rapid Prototyping:** Testing new AI scripts, sensor sweeps, and behavioral changes inside a highly detailed 3D environment without modifying the primary game engine.

Are you planning to build a **specific simulation scenario** (like UAV swarms or ground combat), or do you need help setting up the **C++ development environment** in vsTASKER?

Titan Vanguard Integration

by [VirtualSim](#)

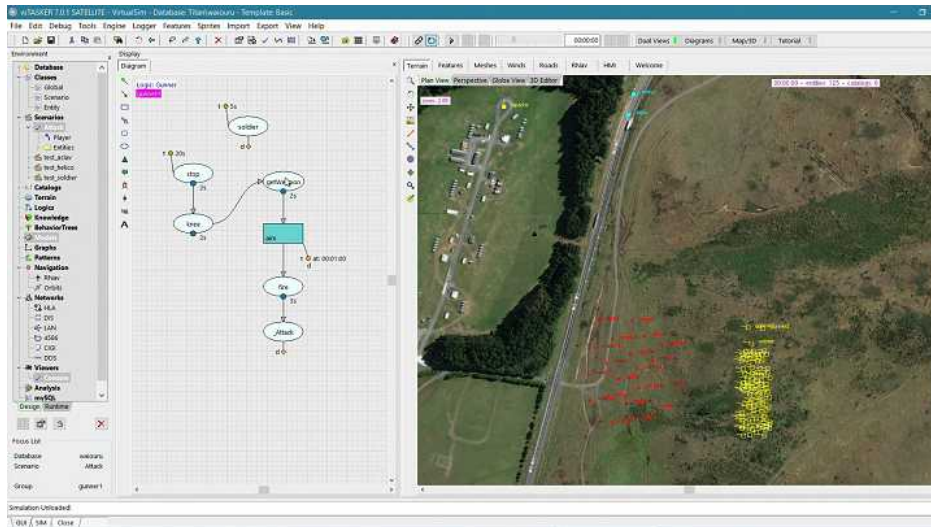
Test demo of a massive scenario including 220 soldiers, 40 mines, 2 gunners and one helicopter, using Titan Vanguard from Calytrix and Titan.IM

The purpose is to demonstrate the capability of vsTASKER engine to be included into the main Titan loop as a DLL host, using directly the C++ API to create and control a scenario at runtime. All the demo has been developed inside vsTASKER, Titan being used as a pure game-engine controlled from its software interface.

In this scripted exercise, all soldiers are directed using waypoints and the helicopter is manually flown using a joystick.

The video below shows the making of, using vsTASKER as a scenario editor.

- Titan Vanguard from Calytrix / Titan.IM
- vsTASKER 7 with Visual Studio 2015
- Gamepad to control helicopter and the camera
- Some ambient sounds have been added to the video
- Computer: Intel Core i5 @ 3.60GHz, 16G RAM, NVidia 1060 GTX 6G



The Titan Vanguard Integration by VirtualSim embeds the vsTASKER logic engine directly into the Titan.IM (Titan Vanguard) 3D simulation core. Utilizing a plug-and-play C++ API, vsTASKER creates a DLL loaded at startup to control all entities and environment behaviors in real-time.

Key Features of the Integration

- **Direct Control:** The vsTASKER Sim DLL uses Titan Vanguard's C++ API to directly command and position objects.
- **Logic Simulation:** vsTASKER handles complex behaviors, rules of engagement, and AI logic, while relying on Titan Vanguard solely for high-fidelity 3D rendering and planetary-scale visualization.
- **Applications:** This combined architecture is purpose-built for battlefield simulations, military training, and commercial exercises.

For a visual demonstration of the vsTASKER and Titan Vanguard battlefield simulation using the C++ API:

TitanIM/Titan Vanguard — The ARMA 3 Killer?!

Would you like to know more about how **vsTASKER's C++ API** handles AI logic for battlefield entities, or explore its capabilities with **other IG rendering engines**?

OS & HARDWARE SUPPORT	INDUSTRY APPLICATION

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