WIND
AN INTEL COMPANY

# Case Study: General Dynamics Uses Wind River Simics to Meet NASA Challenge

## Executive Summary

General Dynamics C4 Systems is a leading integrator of secure communication and information systems and technology. With more than 11,000 employees worldwide, the company specializes in command and control, communications networking, space systems, computing, and information assurance for defense, government, and select commercial customers in the United States and abroad.

General Dynamics was tasked with designing and building NASA's Gamma Ray Large Area Space Telescope (GLAST), now referred to as Fermi Gamma Ray Space Telescope. By simulating the satellite hardware with Wind River Simics, General Dynamics was able to build a Fermi testing and training platform in a more cost-effective, time-efficient manner.

## NASA Challenge

Capable of detecting gamma rays throughout the universe, NASA's Fermi is designed to grant astronomers unprecedented insight into the objects and phenomena that generate this highly energetic form of radiation.

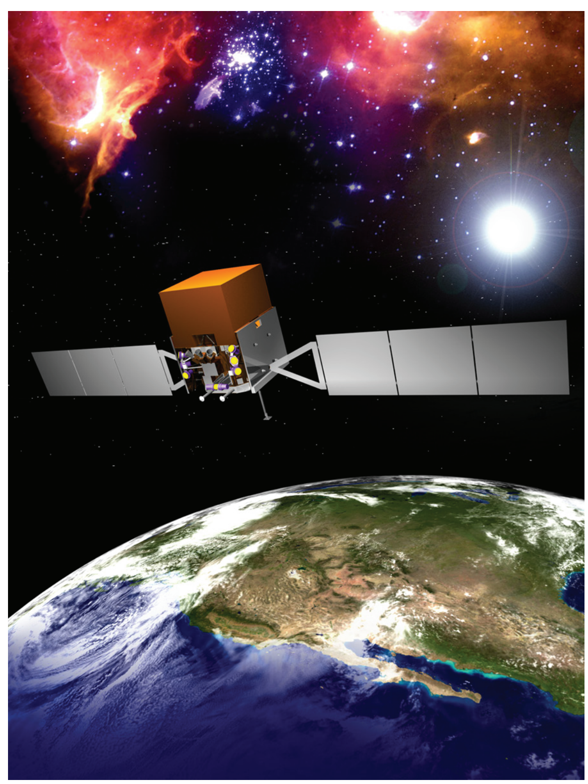
In designing and building Fermi, the General Dynamics C4 Systems business unit was faced with a challenge: It needed to provide an environment that could support rigorous testing by multiple Fermi subsystem groups before the spacecraft hardware was available.

The NASA flight software team required a platform that would allow them to proceed with software development before hardware availability. Development and testing of the embedded flight software generally requires the construction of test models of flight hardware.

While test hardware is a necessary resource for various Fermi subsystem groups, it also constitutes a multimillion dollar expense. With the increased capabilities built into today's spacecraft and the accompanying growth in complexity of the flight software, test models that can support software development have become more difficult, time consuming, and costly to construct.

In addition, General Dynamics needed to furnish a training tool that would enable the Fermi Mission Operations Center (MOC) to prepare for normal spacecraft operations. To develop a routine for day-to-day and contingency procedures, the mission operations team required a training environment where faults could be easily introduced and MOC responses could be implemented and rehearsed well before the spacecraft's launch.

With actual test hardware, failures can be difficult to produce without potential damage to the hardware. It can take hours to create a specific situation on the hot-bench in order to test the response of the operators. But Wind River Simics provides complete control over the system—in particular, injecting hardware faults to train the operators on handling flight anomalies.



### General Dynamics, Wind River Solution

Recognizing the capabilities and advantages of hardware simulation, General Dynamics turned to Wind River and its full system simulator, Wind River Simics. Using Simics, General Dynamics has constructed and delivered a tool that facilitates both software development and MOC training procedures.

Using a virtual model of the hardware, General Dynamics simulates satellite processors and other spacecraft avionics hardware, allowing the embedded flight software to run without detecting the difference from running on real production hardware.

With the robust development environment provided by Wind River Simics, developers can test and debug flight software on a simulated model, minimizing the need for multimillion-dollar test hardware.

Wind River's virtual platform supports the hardware emulation routines implemented by the Fermi MOC. With Wind River Simics, General Dynamics can create a more flexible mission operations training tool that is capable of reflecting failures or irregularities.

According to General Dynamics, the mission operations team can now conduct rigorous training exercises that will aid in recognition of and response to in-orbit failures. Because the hardware simulation can be configured on a PC without any modifications to physical hardware, a range of failures can be easily and repeatedly registered without risk of damage to expensive test hardware.

### Increased Efficiency, Flexibility, Reduced Risk

With Wind River Simics, developers increased control over flight software code for extensive and thorough testing, which contributes to an accelerated software development process as well as higher-quality code.

The use of Wind River simulation technology also reduces the need for expensive test hardware for software development purposes, delivering considerable cost savings.

With the advanced capabilities of Wind River's simulation technology, the mission operations team benefits from a more flexible training environment that enables comprehensive training for mission-critical operations.

In addition, because anomalies can be quickly and easily injected into the simulated environment, the Fermi operations team can undergo extensive testing to ensure that failures are detected and responded to in an efficient and correct manner. This thorough training translates into reduced risk of operational errors once the spacecraft has launched.

With software for mission-critical systems moving in the direction of greater complexity, and scalability becoming essential to meet the needs of multiple subsystem groups, simulation is gaining recognition as a viable and powerful tool for development and training.

Similar to many industries, it is becoming increasingly difficult and pricey for aerospace manufacturers to produce the hardware necessary to support software development, testing, and training. Wind River Simics enabled General Dynamics to create a comprehensive test platform that can easily scale across the whole project to meet the needs of all parties involved while also providing testing capabilities beyond those in real hardware.

### Conclusion

With Wind River Simics, General Dynamics was able to build its testing and training platform in a more cost-effective, time-efficient manner. The Wind River solution gave developers more control over the flight software code, resulting in faster time-to-market and higher-quality code. It also reduced the need for expensive test hardware for software development, enabled a more flexible training environment, and reduced the risk of operational errors once the spacecraft was launched.

WIND
AN INTEL COMPANY