

The Gedae Software Development Platform

 POWERED BY THE IDEA LANGUAGE AND COMPILER



Supercharge Your Software Development Process

No longer does software development dominate cost, schedule and risk of your system development. The Gedae technology solves the core issues of software development by automating the most complex and tedious parts of the development process.

The Gedae Software Development Platform turns software development into a strength by simplifying development, unifying your team and delivering software that is reliable, efficient, reusable and easily maintainable.

Unlike other software development technologies, the Idea Language and Compiler offers:

A New Foundation Built for Multi-Cores/Multi-Processors

The Idea Technology is built on new foundational architecture that extends the existing single core architectures to today's multicore, multiprocessor and complex memory architectures.

Turns Everyone into Parallel Programming Experts

The automation offered by Gedae handles the tedious details of implementing software and refocuses developers on creating powerful, more fully featured software.

Truly Portable Code

With the Idea Language developer describes only the behavior of the software while the Idea Compiler handles the creation of the platform specific code.

Delivers Maximum Performance

The Idea Language and Compiler is not a middleware or layered software solution. The compiler is aware of the unique architecture of your target platform and optimizes the layout of data in memory, data transfers and concurrency. Each compilation of your application produces a tailored version optimized to the platform targeted.

Dramatic Productivity Gains

The benefits enabled by automation compound to deliver dramatic productivity gains.

Proven Capability and Value

Our technologies' use for 40+ production programs is a validation of the functionality and value of the Gedae technology.

At-a-Glance

The Gedae Software Development Platform is a fully featured development environment, powered by the Idea Language and Compiler, with a complete set of tools for implementing, analyzing and tuning applications for multicore, multi-processor and multi-board systems.

Advantages

Software:

- » Maximizes productivity, performance and quality
- » Simplifies development for multi-core and multiprocessor systems

Process:

- » Minimizes schedule and cost
- » Enables reuse of software
- » Facilitates team communication
- » Integrates tool sets

Programmatic:

- » Meet most demanding SWAP-C Requirements
- » Drive refreshes by easily migrating to new more powerful hardware or adding new application features

Organizational:

Turns SW into a strategic advantage by leveraging cumulative advantages

- » Build complete systems with IR&D and business development budgets
- » Bid more, bid more competitively, win more programs

"In the past, tools for increasing software production have often fallen short of their promises. We at EADS feel [Gedae] is a notable exception. Gedae represents a giant leap forward in software production efficiency."

Dr. Elmar Compans
VP Airborne RADAR EADS

The Tools & Tech

The Idea Language

- » General purpose
- » Similar in abstraction to the MATLAB® language
- » Uses familiar syntax and semantics such as algebra, loops, conditionals, functions and data structures

Implementation Controls

- » Partition and Mapping controls allow you to maintain an independent definition of how your algorithm will run on a particular platform
- » The separation enables portability and scalability
- » Optional controls include: transfer methods parameters, function firing granularities & priorities, and partition/map memory

Trace Table

- » Displays the sequence of operations and use of memory
- » Detailed view of execution time, blocking, sends and receives, processor loading and data throughput

Probes

- » Collect output from any function in the application
- » Utilize for regression testing to debug and optimize application

Developing with Idea

1. Develop Your Algorithm

Write your algorithm using the Gedae Idea Language. Use probes to analyze and verify your algorithm with various data analysis and displays, like scopes, spectrograms, 3-D surface and constellation diagrams.

2. Specify your implementation

Once you have developed and verified your algorithm you use the Implementation Controls to define how your algorithm should be implemented on your target platform. First break up your application by grouping your functions using the Partition Table. You then map the Partitions to particular processors using the Mapping Table. An extensive set of controls give you the option of tuning the implementation as much as you need.

3. Compile your application

The Gedae Idea Compiler accepts your algorithm, implementation definition and it's knowledge of your target system and creates a custom optimized version of your application.

4. Analyze

At the user's request the Gedae Idea Compiler collects trace events and probe data. The trace data can be viewed and analyzed with the Trace Table and the associated set of analysis tools. These tools are used to determine which functions are using most time, cost of data transfers and the cause of any processor idle time. The probe data can be analyzed with the Gedae provided tools. You can import .MAT files for comparison or exported in .MAT formats for use with MATLAB®. The probe data is used for regression testing or for verifying implementation correctness.

5. Optimize

Once you've done your analysis, iterate to 'Step 2', adjust the implementation, compile and analyze until you've met your requirements. The automation delivered by the compiler makes this a very fast process that you can loop through dozens of times a day.

6. Finalize and Deploy

Once your application is finalized you can deploy it as either a standalone application or as a callable library function that can be easily integrated into existing applications.



Find Out More.

Learn more about how Gedae can help you supercharge your software development by visiting www.gedae.com.

Supported Platforms

Processor Types:	32/64 x86, DSP, PPC
Processor Vendors:	Intel, AMD, TI, Freescale
Processor Cores:	1 to n...
Processors / Board:	1 to n...
Boards / System:	1 to n...
Board Vendors:	Curtis-Wright, CSPI, GE, Kontron, IBM, Intel based platforms from a variety of vendors
Embedded OS's:	VxWorks, Linux, Sys/BIOS

Requirements

Windows Operating System
Visual C ++ v.6.0 or later
Exceed / Exceed XDK
Linux Operating System
LessTif or Open Motif
X Server
Ansi C compiler (GCC preferred)