

ATEASY 4.0

ATEASY FEATURES

- 32-bit object-oriented programming environment framework supports Windows 9x/ME and NT/2000/XP
- Rapid Application Development (RAD) environment enables short development cycles
- Instrumentation/Test System like modular structure allows easier development and maintenance
- Built-in Application Builder generates royalty-free run-time executables (EXE files)
- Built-in Test Executive and Profile editor offers full control of test execution, sequence, and conditions
- User-defined, plain language commands offer program legibility for easier maintenance
- Automatic generation of HTML and text-based test logs instantly display test results
- Built-in Configuration Management tools allow easier project management
- Connectivity with Microsoft Source Safe enables direct check in, out, add, and compare files
- Multithreading support allows simultaneous execution of multiple code segments
- Visual Basic-like Form Editor generates graphical user interfaces, menus, and controls
- Event-driven programming support facilitates more efficient program execution
- Supports ActiveX, OCX, OLE, DDE, DLL
- Vendor-independent open-system architecture supports PXI, PCI/cPCI/PXI, VXI, GPIB, TCP/IP, RS-232, ISA, and more
- Supports VXI Plug-and-Play, Function Panel, and IVI drivers

ATEasy

- ✓ Instrument Independent
- ✓ Test Plan Oriented
- ✓ Scalable



EE
Evolution Engineering
2002
READERS'
CHOICE



WHAT IS ATEASY?

ATEasy is a rapid application development framework for functional test, ATE, data acquisition, process control, and instrumentation systems. ATEasy provides all the necessary tools to develop and maintain software components, from instrument drivers to complex test programs. It is designed to support and simplify ATE projects with long life cycles. With ATEasy, test applications are faster to generate and easier to maintain. ATEasy includes a complete test development suite and a test executive specifically designed for test applications. The ATEasy development environment combines the ease of Microsoft Visual Basic and the flexibility of Microsoft Visual C++; a complete object-oriented, 32-bit Windows programming environment.

INDUSTRIES AND APPLICATIONS

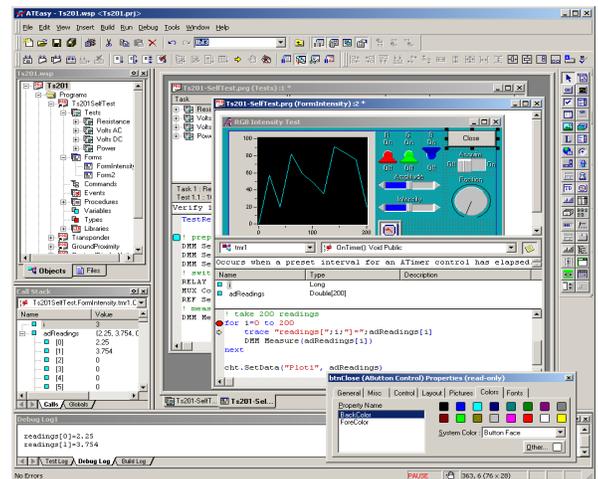
ATEasy is used successfully in the following industries and applications:

- Functional Test Systems
- Instrumentation Systems
- Avionics
- Aerospace
- Military
- Medical
- Wireless Telecom
- Semiconductor
- Automotive
- Electro-Optical
- Process Control

WHY USE ATEASY?

INTEGRATED DEVELOPMENT FRAMEWORK

Test engineers build their test application from components modeled after real-world test systems. These components include a System, Drivers, Programs, Tests, Commands, and more. ATEasy provides a streamlined, easy-to-follow framework that directs the user to create re-usable components. The result is a test application that is faster to generate and easier to maintain.



The ATEasy IDE

Geotest
Marvin Test Systems, Inc.

www.geotestinc.com

ATEASY 4.0 TEST DEVELOPMENT ENVIRONMENT

SHORT LEARNING CURVE

The ATEasy development environment provides a familiar graphical user interface that allows Microsoft Visual Basic™ or Visual C++™ users to feel right at home. First time users can use the Application Wizard to generate applications quickly. Application components are displayed in an easy-to-browse tree view that serves as a basis for the application specific components.

RAPID APPLICATION DEVELOPMENT (RAD)

ATEasy's application framework consists of well-organized components that allow engineers to partition and organize their test code during development. When debugging and validating, engineers can use these components to quickly isolate problems. Once a modification is applied, smaller portions of the application code can be executed independently without running the entire application, which provides rapid and short development cycles.

ATEASY COMPILER

The ATEasy compiler is *fast*. It is so fast that you'll hardly see it compiling. During debugging, ATEasy's Just-In-Time compiler compiles only the necessary code as required. Once debugging is complete, the **Build** command creates an executable file.

EASIER TO PROGRAM

ATEasy applications can be created using menus or by typing commands directly. With menu commands, users can insert driver commands, procedure calls, and even flow control statements with a few clicks of the mouse. If the users choose to type, ATEasy's code completion tools

provide suggestions on completing the unfinished statements. Code completion tools include Parameter suggestion and Command completion.

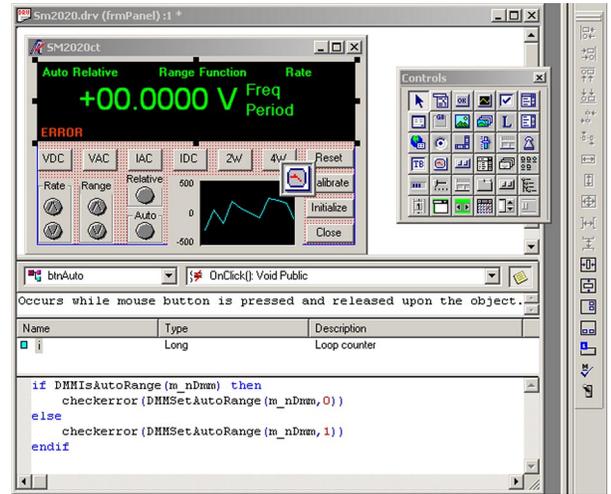
ATEasy also provides tool tips that describe the function call and arguments to the parameter that you are about to type. ATEasy displays information about symbols (variables, procedures, commands, etc.) when users hover over the symbol with a mouse. Right-clicking the symbol, then selecting **Go to Definition** opens a window that allows you to view or edit the symbol. After modifying your code, you can select and click on **CheckIt!** to verify that the code is correct.

USER INTERFACE PROGRAMMING

ATEasy's Form Editor is very similar to the Microsoft Visual Basic™ form editor, providing similar sets of forms types, menus, controls, and event programming. ATEasy's forms can accommodate any of your system-installed ActiveX controls, providing you with the extensibility and flexibility to use hundreds of ActiveX controls and components available from third party vendors.

UNIQUE USER INTERFACE AND TEST APPLICATION PROGRAMMING MODEL

ATEasy provides a unique event-driven approach to user interface programming for test applications. While a form is displayed to provide user interface to the test application, ATEasy test programs run in parallel, relieving you from "pumping" windows periodically as you would do in conventional user-interface programming. When the user clicks a button or selects a menu, an event is generated that interrupts your test program to handle and execute the event code. For time-critical applications, forms can be created in separate threads; this causes your tests to run 'isolated' from your user interface events, with little or no effect on test timing.



Quickly and easily build user forms in ATEasy

MULTITHREADING

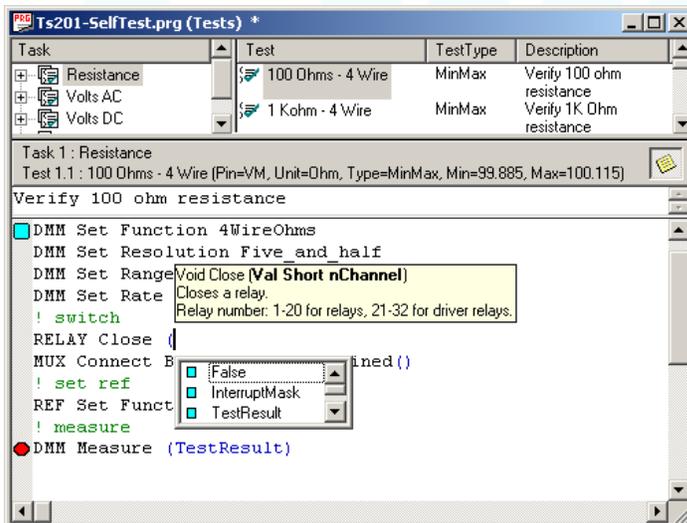
ATEasy provides full support for Windows' multithreading model, which allows users to execute multiple code segments simultaneously. Synchronization objects such as semaphore and events allow users to synchronize thread execution to protect the application's resources from re-entrance. ATEasy's robust multithreading model lets users use any user-interface objects or ActiveX controls from any thread without any special programming.

EXCEPTION AND ERROR HANDLING

ATEasy provides a unique approach to error and exception handling. Errors are generated by the **error** statement, run-time, or instruments communication. Once an error occurs it can be trapped and handled locally using the **try-catch** statement or at the module level using **OnError** module event. These minimize the developer's efforts on placing error-handling code throughout the application. When an error is generated, the application can decide whether to **ignore** or **retry** the statement that caused the error. It can also abort the application, reset, or re-throw the error, so other modules or ATEasy's default error handling can handle the error.

INSTRUMENT DRIVERS

ATEasy supports thousands of VXI Plug-and-Play, Function Panel or IVI instrument drivers from vendors such as Agilent Technologies and National Instruments in addition to its own provided drivers. ATEasy also provides various tools for rapid creation and debugging of instrument drivers. These tools include



ATEasy's code completion tools speeds test application development

ATEASY 4.0 TEST DEVELOPMENT ENVIRONMENT

I/O Tables that contain steps used to communicate and control instruments over various bus types (e.g., GPIB). ATEasy's internal library also contains many functions that can control instruments (e.g., GPIB, RS232, TCP/IP, VXI, PC IO, memory functions, and more). A communication monitor window shows the data sent or received from the instrument and can be used to identify and debug these instrument drivers.

FLEXIBILITY

ATEasy fully supports Microsoft's Component Object Model (COM), DLL, and DDE. This open system architecture enhances your application with ActiveX controls, spreadsheets, databases, word processors, web browsers, and more.

SELF-DOCUMENTING

ATEasy's user-defined commands allow users to create plain English-like Command statements, such as:

```
DMM Set Function VDC
DMM Measure (TestResult)
```

Statements such as these can control instruments or test systems. The resulting test code resembles the Test Requirement Documentation (TRD). ATLAS language users will find Command statements similar to ATLAS statements.

INTEGRATION WITH MICROSOFT SOURCE SAFE

ATEasy provides seamless integration with Microsoft Source Safe. You can **Check In**, **Check Out**, or **Add Files** directly from ATEasy to Source Safe. You can even compare different versions of any ATEasy file to see what, when, where, and by whom files were modified.

SOURCE LEVEL DEBUGGING

The ATEasy debugger provides the versatility of conventional software development tools. For example, while the

application is paused, users can **Step In**, **Step Out**, and **Step Over** when executing code. Users can set breakpoints, **Run to Cursor**, and/or **Set the Next Statement** from which to run. ATEasy has windows that show Call Stack, Local Variables, and

Watch; while hovering on variable in the code window, ATEasy displays its value. ATEasy also has specialized instrument control debugging features such as the **Communication Monitor** that shows you what was sent or received from the bus to control the instrument. Other commands such as **Doit!** and **Loopit!** let execute selected or typed code to verify driver commands or procedures without executing the entire application.

TEST PROGRAMS DEBUGGING

The ATEasy debugger can execute a Test or a Task without executing the whole application, saving you the time needed to run the whole application until the required test is reached. Additional test program debugging features include: **Skip Task/Test**, **Set Next Task/Test**, **Run to Task/Test**, repeat **Current Task/Test**, **Taskit!**, **Testit!**, and run conditions such as: **Continuous**, **Task By Task**, **Test By Test**, **Stop on Failure**, and more.

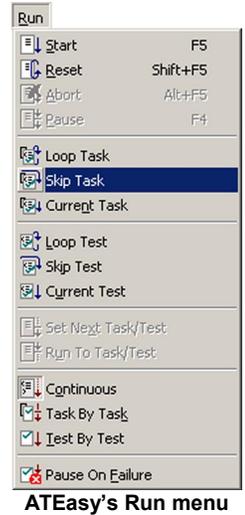
BUILT-IN CUSTOMIZABLE TEST EXECUTIVE

ATEasy comes with a ready-to-run Test Executive module. Once the Test Executive is used, the application has a complete test executive user interface that allows users to select and run a test program, select test to include or exclude from running, debug, view, and print test logs. The Test Executive module can be customized to accommodate any requirements. This plug-in architecture allows organizations to re-use the customized test-executive they have developed.

RUN TESTS IN ANY SEQUENCE

A Profile plug-in module allows you to create profile files to define custom testing sequences. Each profile contains a list of steps that specify the program, Task, or Test to run from the application and the number of times the step should be executed.

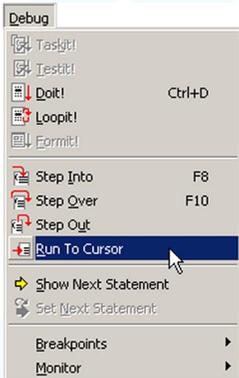
The profile is fully integrated with the Test Executive module and can alternatively be used as a stand-alone module.



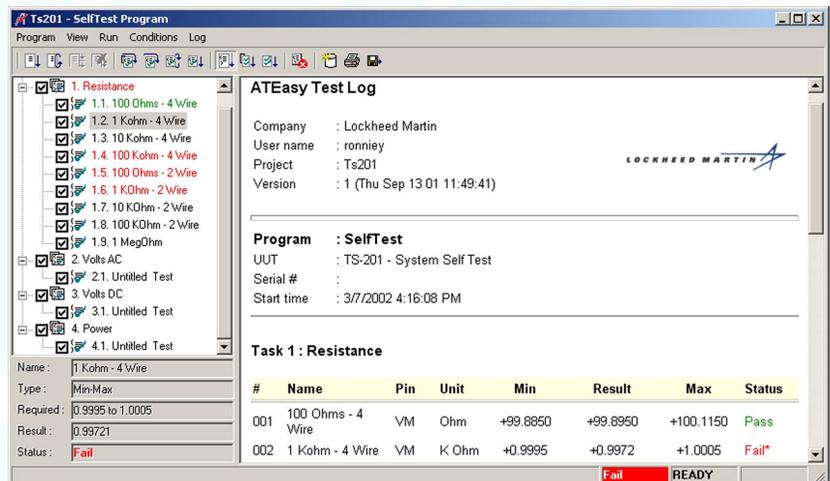
ATEasy's Run menu

TEST DEVELOPMENT CHALLENGES

Test application development demands an environment that is interactive, rapid, and yet flexible. While conventional software development requires simple debugging, Automatic Test Equipment (ATE) related applications require interactive and specialized debugging and integration that result from software, hardware, timing, and communication considerations. Upon completion of the test application, regular maintenance requires skilled engineers who are familiar with both the application and the Unit Under Test. Companies need to consider the development costs and the life-cycle costs of a test application. Having the right tools in an ATE software development environment is therefore the most important factor. A poorly selected



ATEasy's Debug menu



ATEasy's Test Executive

ATEASY 4.0 TEST DEVELOPMENT ENVIRONMENT

ATE software package may result in a long and expensive development cycle.

WHAT ARE ATEASY MODULES

ATEasy applications are created from a project file that contains the application modules files. ATEasy modules are modeled after a test system. Three types of modules are available:

- **Program Module** – contains the application
- **TestsSystem Module** – lists the application drivers and their configuration. For example: a GPIB instrument driver configuration may have its address, terminator, etc.
- **Driver Module** – contains the commands and functions that are required to operate an instrument as well as the interfaces it supports (e.g., GPIB, RS232, etc).

Each ATEasy module contains sub-modules such as Forms (for user interface), Commands, Procedures, Variables, Types and Libraries.

ATEASY TESTS AND TASKS

The Program module contains the necessary tests required to test a UUT. A Task consists of a group of Tests, which test the same block or logical unit in the UUT. Each Test contains code and various properties such as its Name, Type, Pin, Unit, Result, Status, and more. Various test types are built-in, such as Min-Max, where the result must fall within required Minimum and Maximum values.

The test code is responsible of taking measurements and assigning them to TestResult, an internal ATEasy variable. When the test code is executed, TestResult is used to calculate the test status: Pass or Fail. The test status, together with the test information, is used to automatically generate the test log.

WHAT OUR CLIENTS HAVE SAID...

ATEasy has enabled me to put together what I wanted with enough growth room for years to come.

*Lowell Parsons,
Test Engineer,
Litton Laser Systems*

We've been using ATEasy for quite a while now. I have to say it's the easiest environment we've ever used.

*Jim McBride,
Test Engineer,
LXE Inc.*

Everybody at Interstate has agreed that ATEasy by Geotest is head and shoulders above everyone else.

*William McGraw, Jr.,
Staff Test Engineer,
Interstate Electronics Corp.*

SPECIFICATIONS

GENERAL DESCRIPTION

- 32-bit Integrated Development Environment for Windows 95/98/ME, Windows NT 4.0, 2000, and XP
- Generate royalty free 32-bit Windows executable files (EXE)
- Provide a framework for test application and for instrument control
- Generate Workspace, Project, Program, System, and Driver files
- Files can be saved and loaded in text or binary format
- Tools to compare files and versions including Microsoft Source Safe Integration and built in versioning to each file type
- Generate customizable test reports in HTML or text format
- Generate documentation automatically
- Multi-level user security
- Wizard for generating applications

EDITING

- Drag/Drop objects
- Cut/Copy/Paste/Delete
- Find/replace for object's name, description, or code across multiple modules
- Merge objects
- Undo/Redo (unlimited)
- Context menu for all objects

EDITING WINDOWS

- Build Log
- Workspace window
- Test Log (HTML or Text)
- Properties window with multi-tab (modeless)
- Debug Log
- Variables window
- Workbook user interface model
- MDI Document windows with Tree/Object views

CODE GENERATION/EDITOR

- Syntax Highlighting
- Insert Flow Control Menus
- Auto Type Information
- Auto Parameter Information
- Auto List Commands
- Bookmarks
- Auto List Members
- Insert Commands cascading menus
- Insert Symbol Dialog
- Auto Parameter Suggestion
- Font, Tabs, Auto Indent

FORM

- Drawing
- Event programming
- Third Party ActiveX controls
- Menu Bar and Context Menus
- Tool tip
- Win Help/Html Help
- Procedures, Variables, Events
- MDI Frame, MDI Child, and Normal Forms

FORMS

- Customization
- Toolbars
- Tools menu
- Keyboard keys
- Menu
- Docked/Float/MDI windows

FORM EDITOR

- Align controls
- Test Form
- Grid
- Space evenly controls
- Auto arrange controls
- Check duplicate mnemonics
- Same size controls
- Lock Controls
- Margins
- Center across the form
- Tab Order controls

COMPILER

- Checkit!
- Check All
- Re-Build
- Stop Build
- Check Module
- Build
- Show Error
- Execute

TEST LEVEL DEBUGGING

- Loop Task/Test
- Current Task/test
- Run to Task/Test
- Stop on Failure
- Continuous/Task By Task/Test By Test
- Skip Task/Test
- Set Next Task/Test
- Testit!
- Taskit!

SOURCE SAFE INTEGRATION

- Check In / Out
- Compare Files
- Show Status / Properties
- Add Files / Project
- Show History

SOURCE LEVEL DEBUGGING

- Reset
- Pause/Continue
- Step Over
- Breakpoints
- Loopit!
- Run to Cursor
- Show Next Statement
- Abort
- Step Into
- Step Out
- Doit!
- Formit!
- Set Next Statement

DEBUG WINDOWS

- Calls Stack/Locals with a separate tab for module variables and change value
- Watch window with expression evaluator
- Debug window for executing code in immediate mode without adding the code to a test program
- Monitor (communication) window, with interface source filtering

ATEASY 4.0 TEST DEVELOPMENT ENVIRONMENT

INSTRUMENTATION INTERFACES

- GPIB: Computer Boards, Keithley/MetraByte/CEC, Agilent/HP, National Instruments
- VXI: National Instruments – MXI
- COMM: Any Windows compatible serial and infrared port
- File/Device
- WinSock: Client/Server, UDP/TCP
- ISA and PC-based instruments

STATEMENTS

- If - Else
- For Next
- While
- Exitloop
- Run
- Reset
- Test
- ExitTest
- Ignore
- Abort
- Try - Catch
- Select - Case
- Repeat - Until
- Loop
- Continue
- Exit
- Task
- ExitTask
- Retry
- Pause
- Error
- Print

DATA TYPES

- Char
- Short
- Word
- Long
- Float
- Currency
- String
- Variant
- Object
- Enum
- Byte
- Bool
- WChar
- DWord
- Double
- DateTime
- BString
- Procedure
- Structure
- Typedef

COM

- Load Type Libraries
- Display type libraries contents
- Early and Late Binding
- Objects can be use from any thread
- CreateObject/GetObject

CONTROLS (ACTIVEX)

- AButton
- ACheckBox
- AGroupBox
- Almage
- ALabel
- ALog
- AScrollBar
- AStatusBar
- ATextBox
- AToolBar
- AChart
- AComboBox
- ACommon-Dialog
- AlmageList
- AListBox
- ARadioButton
- ASlider
- ASwitch
- ATimer

OPTIONS

- Directories
- Log
- Text Editor
- Interfaces
- Tests
- Workspace

UI CLASSES

- AAxis
- AControl
- AForm
- AMenu
- APlot
- AThumb
- AClipboard
- AFont
- AlmageList-Image
- APicture
- AStatusBar-Pane
- AToolBar-Button

CLASSES

- App
- ADriver
- AMutex
- ASemaphore
- ATask
- AUser
- ACriticalSection
- AEvent
- AProgram
- ASystem
- ATest
- Ausers

PROCEDURE GROUPS

- ActiveX
- DDE
- File System
- GPIB
- Math
- Misc
- VXI
- Variants
- Interrupts
- WinSock
- Driver/Info
- Interfaces
- Serial Communication
- Log File Information
- Multi-Threading
- String Manipulation
- Port/Memory I/O

HELP/DOCUMENTATION

- Getting Started manual
- User's Guide manual (on-line)
- Reference/Programming Language manual (on-line)
- HTML based help with TOC, Index, Search
- Context sensitive help for dialogs and keyword
- What's this help

OTHER

- DLLs
- Module events
- Multi-Threading
- Exception Handling (try-catch, error, OnError)

ORDERING INFORMATION

GT8000-xx	ATEasy Single User Software Package
GT8003-xx	ATEasy with a 1-year Subscription and Support and 3-day training at Geotest
GT8320-xx	ATEasy with a 1-year Subscription and Support
GT8320-2Y-xx	ATEasy with a 2-year Subscription and Support
GT8000-UG-xx	ATEasy Upgrade (from 3.0 to 4.0) without a Subscription and Support Agreement
GT8021-UG-xx	ATEasy Upgrade (from 3.0 to 4.0) with One-Year Subscription and Support Agreement
GT8022-UG-xx	ATEasy Upgrade (from 3.0 to 4.0) with Two-Years Subscription and Support Agreement

NOTES

Replace -xx with "-SK" for Software Key, "-KP" for Parallel Key, "-KU" for USB Key, "-NL" for Network License

Marvin Test Systems, Inc.