

# ATEasy



## TEST EXECUTIVE AND DEVELOPMENT STUDIO

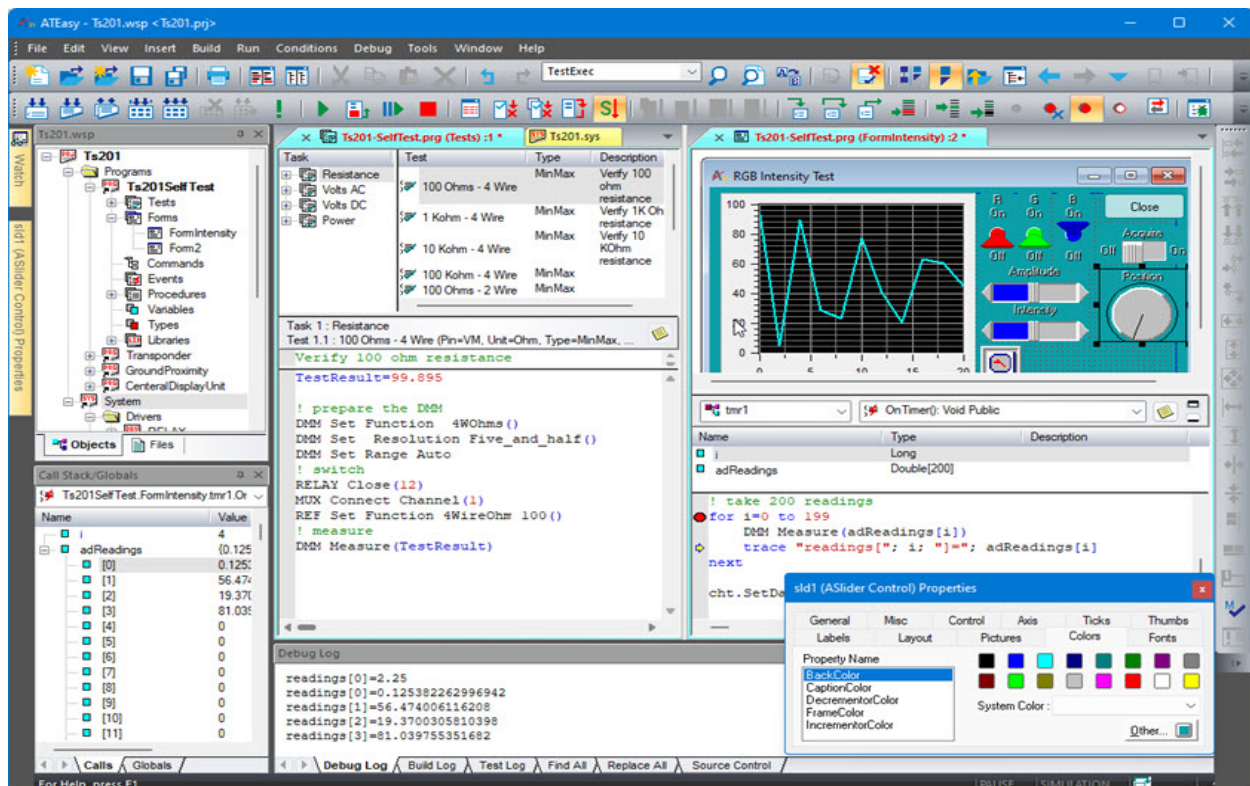
- Software development environment and integrated, customizable Test Executive for execution, sequencing, debugging and fault analysis of tests
- Comprehensive simulation capabilities accelerates test program development and deployment
- User interface generation including form editor, event programming, menus, and controls
- Open architecture supports external software and hardware interfaces: DLL and C header files, .NET, ActiveX, LabVIEW, Function Panel Drivers, IVI, GPIB, LXI (TCP/IP), VXI, USB, Serial and more
- Built-in Application Builder generates royalty-free run-time executables (exe files), and libraries (dll files)
- 32 and 64-bit Application Bitness running on Windows XP-Windows 11 bundled to one setup
- Cost-effective version (ATEasy-Lite) for OEM applications

**ATEasy®**  
Test Executive And Development Studio



## DESCRIPTION

ATEasy is a test executive and a rapid application development framework for functional test, ATE, data acquisition, process control, and instrumentation systems. ATEasy provides all the necessary tools to develop, deploy and maintain software components - including instrument drivers, test programs, and user interfaces, as well as a complete and customizable test executive. It is designed to support and simplify ATE system applications with long product life cycles. With ATEasy, test applications are faster to generate and easier to maintain.



ATEasy IDE (Integrated Development Environment)





ATEasy's open architecture provides easy to use access to many industry standard hardware and software interfaces including GPIB, VXI, PCI/PXI, USB, LXI/TCP-IP, Serial Communication, DLLs, ActiveX, .NET assemblies, HTML, VXI Plug&Play Function Panel drivers, IVI drivers, LabView VIs, C Header files and more.

For OEM suppliers of test systems, ATEasy-Lite offers a cost-effective, full-featured, test executive and test development software suite that can be bundled with the supplier's ATE systems. ATEasy-Lite includes ATEasy's test executive driver but only supports two non-OEM drivers. Additionally, ATEasy-Lite does not offer ".exe" file generation capability. Licensing is software based, with the OEM's hardware providing the license key. Note that the OEM hardware must always be connected for use with this version of ATEasy. ATEasy-Lite can be upgraded to ATEasy with the purchase of an upgrade option.

To learn more about ATEasy click on the other ATEasy tabs/pages in this section or check out the [Tutorials/Videos page](#).

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's Visual C++, object-oriented, Windows programming environment.

## WHY ATEASY?

ATEasy provides the test engineer with a multitude of benefits, including:

- **Integrated Framework**

ATEasy provides a streamlined, easy-to-follow framework that directs the user to create re-usable components modeled after real-world test systems. Components include a System, Drivers, Programs, Tests, Commands and more.

- **Re-usability and Scalability**

The ATEasy framework provides users with the ability to create re-usable software components such as instrument drivers, system components and test programs. These components can be reused from system to system reducing the overall cost of creating and maintaining an application.

- **Test Executive**

ATEasy provides a royalty free Test Executive. User group customization and assignments of privileges is easily done with little or no programming. All the tools to execute, create datalogs, and debug your tests are included. Additional modules are included that provide test sequencing and fault analysis.

- **Short Learning Curve**

ATEasy 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. The ATEasy application framework and modular structure offer placeholders for your code reducing the learning curve.

- **Rapid Application Development (RAD)**

ATEasy offers reduced program development cycles for coding, executing and debugging of your code. During debugging, ATEasy's Just-In-Time compiler compiles only the necessary code as required, supporting fast development cycles. ATEasy allows you to write several lines of code, highlight them and then click on the DoIt! button - allowing the execution of only the highlighted code without running the complete application.

- **Fast and Easy Programming**

Applications can be created via menu commands or by typing. Commands can be combined with ATEasy language statements such as if, repeat, etc. When typing, ATEasy's code completion tools provide suggestions for completing the unfinished statements, and an integrated code verification utility, CheckIt! provides immediate code verification for syntax errors. Application wizards help you create a framework for your test application and ATEasy's modular structure allows engineers to partition, organize, locate and re-use the test code. A built-in code browser offers immediate access to symbols such as functions or variable definitions. The ATEasy internal library includes many classes and functions providing support for your software components such as drivers, tests, forms and controls, and for basic operations such as math, string manipulation, communication, etc.

- **Multi threading Support**

ATEasy provides full support for the Windows multi threading model, which allows users to execute multiple code segments simultaneously. Synchronizing objects such as semaphores and events allows users to synchronize thread execution to protect your application resources from re-entrance. ATEasy's robust multi threading model lets users use any user-interface objects and ActiveX controls from any thread without any special programming, unlike other conventional programming environments.



- **Instrument Independence**

Thanks to ATEasy's exclusive Instrument Interchangeability Technology (I2T), test engineers can easily define plain language driver commands for instruments types such as a DMM, ARB, etc. The commands can be implemented differently for different instruments of the same type. Commands can be linked to DLLs, ATEasy procedures, etc. For test programs, these commands are independent of the implementation, and interface type (GPIB, VXI, etc). Consequently, when instruments need to be replaced, a new instrument driver may be used while the test programs remain unchanged.

- **Test and Source Level Debugger**

The ATEasy debugger provides the features found in conventional software development tools such as; Step In, Step Out, Step Over, Breakpoints, Run to Cursor, Set the Next Statement to run, and more. You can create and watch variables and expression values, execute code in a debug window, view a call stack, and view a running thread. In addition, ATEasy includes test level debugging tools allowing you to select and run tests, loop on tests, skip tests, pause on a failed test, and repeat a test.

- **Self Documenting**

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

```
DMM Set Function VDC
RELAY Close (1)
DMM Measure (TestResult)
```

The resulting test code resembles a Test Requirement Document (TRD), which is easy to understand and is instrument independent. System and Program level commands allows further abstraction of your test system and the unit under test:

```
Program Set Engine RPM (4000)
System Counter Measure (1, TestResult)
```

- **Open System Architecture**

ATEasy supports many hardware and software standards including: GPIB, Serial communication, VXI, USB, PCI/ISA, LXI (TCP/IP), DLL, COM/ActiveX controls, .NET assemblies and Controls, LabView VI/LLB files, VXI Plug&Play Function Panel drivers, I/O drivers, the importing of C header files, Microsoft SourceSafe and more. By supporting a wide range of standards, you can extend and enhance ATEasy's built-in functionality with spreadsheets, databases, word processors, web browsers and more..

- **Enterprise and Multi-User Features**

- Support for Source and Version control software (TFS, SVN, GIT, Source Safe and more).
- Side by Side comparison of project folders, sources and merge capability.
- Reusable components architecture provide a simple way to standardize across the enterprise.
- Automatic records of changes generation.
- Encrypt and protect sources to protect your investment or code from changes.
- Multi-user IDE and Test executive with user groups and privilege based access to commands and features.

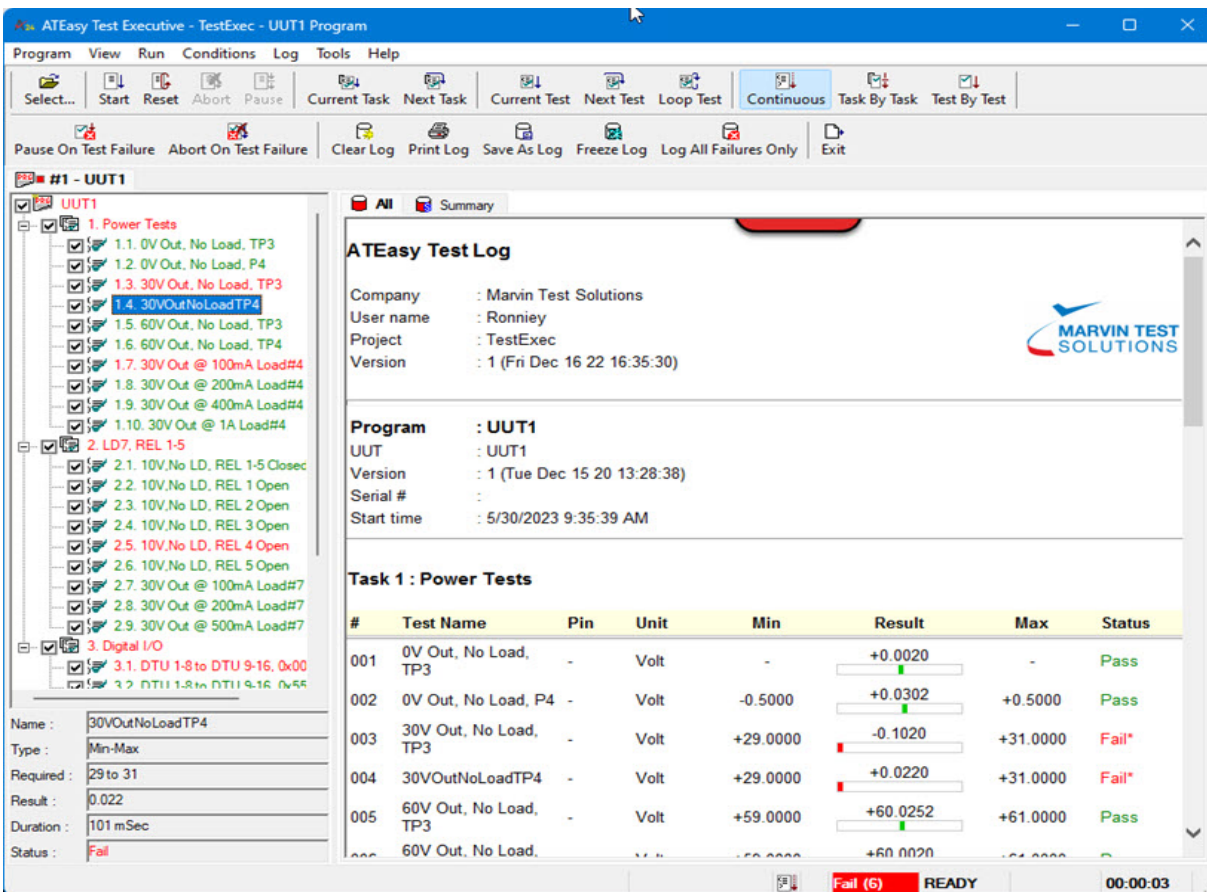
- **Compatibility and Deployment**

- Worry free upgrade to the latest ATEasy supporting backward compatible of ATEasy sources and executable allowing you to upgrade with no code changes.
- Side by side installation allowing you to install and use multiple ATEasy versions on the same machine.
- 32 and 64 Application Bitness are installed to generate, run and debug 32 or 64-bit Applications.
- Small footprint setup for fast deployment.
- Running on Windows XP (sp3) to the latest Windows 11.



## TEST EXECUTIVE

ATEasy is provided with a royalty-free test executive that can be deployed on any number of test systems. Once the test executive is inserted to your project, your application has a complete test executive user interface that allows selecting and running a test program, debugging, viewing and printing test logs. Typically ATEasy tests are written with the ATEasy language however you can extend ATEasy's functionality and execute any external code or library including .NET assemblies, DLLs, LabWindows/CVi or function panels functions, LabView VIs and more. Plug-in modules can also be added to the test executive for added functionality.



ATEasy Test Executive

## CUSTOMIZATION AND USER INTERFACE

ATEasy's test executive architecture is easy to customize. Using the Customize window the user can change the test executive layout, menus, toolbars, touch panel interface, options and more. Programming commands can further extend this capability. Multiple users and user groups enhance customization and provide a way for the administrator to limit the test executive's features for certain users across the network.

## PROFILE AND TEST SEQUENCES

A plug-in Profile module can be used by the test executive allowing the user to create, save and run test sequences. A Profile editor is provided to create the test sequence and set the actions between tests and tasks (group of tests) as required based on the test results.

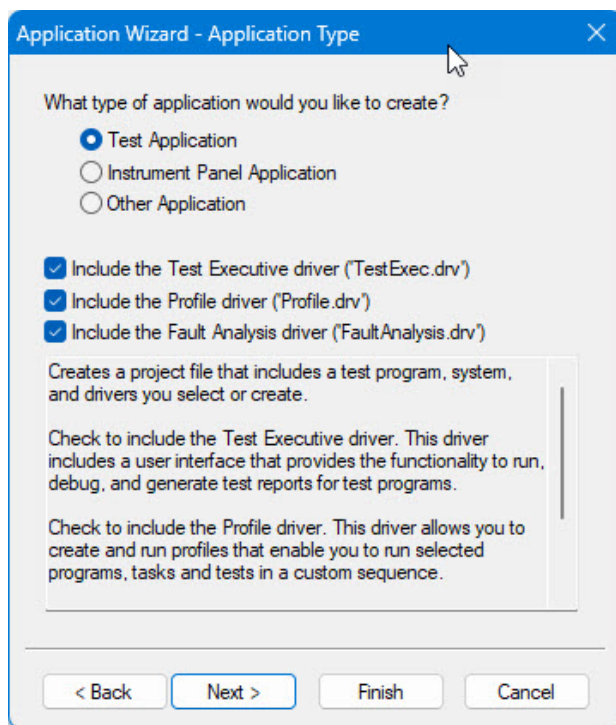
## FAULT ANALYSIS

A plug-in Fault Analysis module can be inserted into your project providing a powerful troubleshooting tool to debug and fix the UUT. The Fault Analysis module analyzes your test results and recommends to the operator how to fix the UUT based on test conditions set met by the test results. A Conditions editor is provided to create the test result patterns and required repair action(s).



## ATEASY FRAMEWORK

ATEasy's development environment is modeled after typical ATE applications. The development environment includes modules in a hierarchical and layered structure as shown in the tree view. Engineers can use the pre-defined structure as a template to partition and develop a test program. During debugging, integration, or maintenance, this modular structure simplifies the process by isolating problems. First time users can use the Application Wizard to generate applications quickly. Application framework components are displayed in an easy-to-browse tree view that serves as a basis for the application specific components.



ATEasy Application Wizard

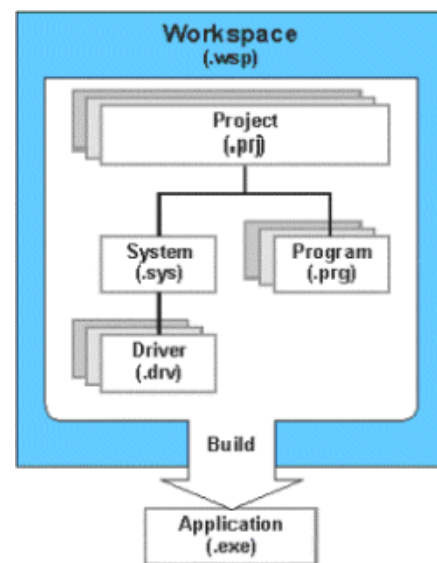
## ATEASY MODULES

An ATEasy application is created from a project file that lists the application modules files, which are modeled after real-life test systems. These modules allow test engineers to quickly prototype, design, and modify test systems. ATEasy provides three types of modules:

**Program Module:** The Program Module is UUT specific. The Program module contains the necessary tests required to test a UUT. The Program Module follows the guidelines of a Test Requirement Document (TRD) or Test Plan and is divided into Tasks and Tests sub-modules. ATEasy Workspace Block Diagram

**System Module:** The System Module is Test system specific. The System module contains the hardware configuration of a given test system. It reflects the currently installed instruments. For example: A GPIB instrument driver configuration includes its address, terminator, etc.

**Driver Module:** The Driver Module is Instrument specific. The Driver module contains the commands and functions that are required to operate an instrument as well as the interfaces it supports (for example, GPIB, RS232, etc).



ATEasy's Tree Structure

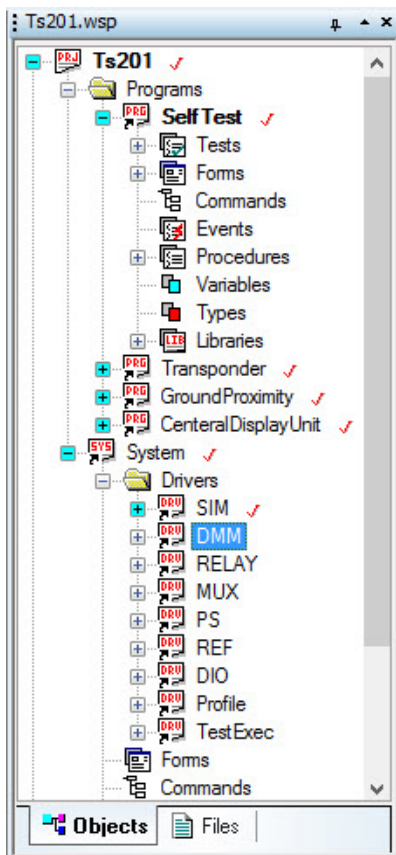
## ATEASY SUB-MODULES

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

- **Task/Test Sub-Module:** The Program module contains the necessary tests required to test a UUT. A Task sub-module includes a group of Test sub-modules, which test a block circuitry or logical unit of the Unit Under Test. Each Test sub-module 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 a required Minimum and Maximum value. The test code is responsible of taking measurements and assigning it to a Test Result, an internal ATEasy variable. When the test code is executed, the TestResult will be used to calculate the test status: Pass or Fail and together with the test information the test log will be automatically generated.



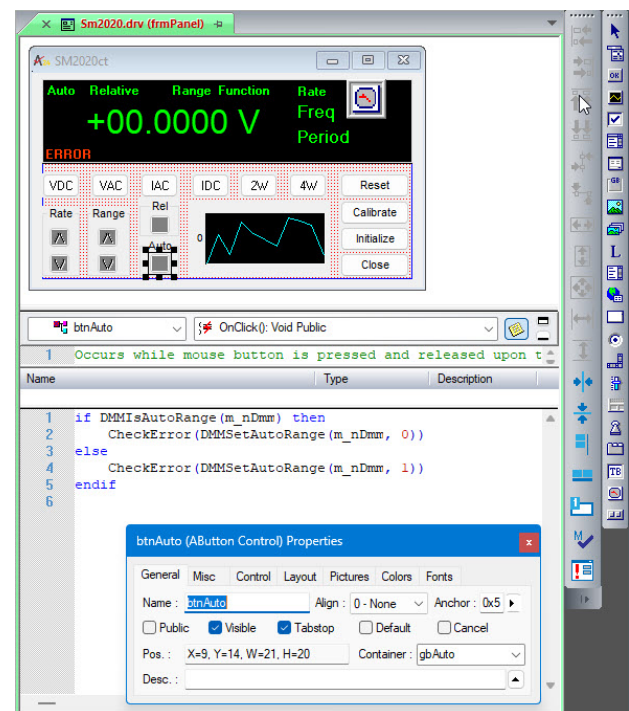
- **Commands Sub-Module:** ATEasy's "command sub-modules" are under the Driver, System, and Program modules. ATEasy's exclusive Test Engineering Language (TEL) is the individual user-defined commands (see figure shown) in the command sub-modules. Calling attached procedures (DLLs or user-defined functions), the TEL commands appear in cascading menus in the ATEasy menu bar. The users can easily insert them into Test sub-modules instead of entering manually.
- **Other Sub-Modules:** System, Driver, and Program modules contain additional "sub-modules", serving as containers for objects such as Forms, Commands, Events, Procedures, Variables, Types, and Library. The test code is "partitioned" into the appropriate sub-modules depending on whether it is UUT specific (Program), system specific (System), or instrument specific (Driver).



ATEasy Workspace Window

## GRAPHICAL USER INTERFACE (GUI) PROGRAMMING

ATEasy's Form Editor is very similar to Microsoft's Visual Basic form editor - providing a powerful and feature rich GUI programming environment for constructing interactive instrument control and test program status / control. Like Visual Basic, ATEasy provides similar sets of forms, types, menus, controls, and event programming. In addition, ATEasy's forms can accommodate hundreds of ActiveX controls and components from third party vendors.



Form Editor Window

## SOURCE CONTROL INTEGRATION

ATEasy provides seamless integration with various source and version control providers such as Microsoft's Source Safe, Team Foundation Server, Git, Subversion and more - providing a robust configuration management environment for managing test programs. You can Check In, Check Out, or Add Files directly from ATEasy to your source control repository. You can even compare and merge different versions of any ATEasy file to see what, when, where, and by whom files were modified.

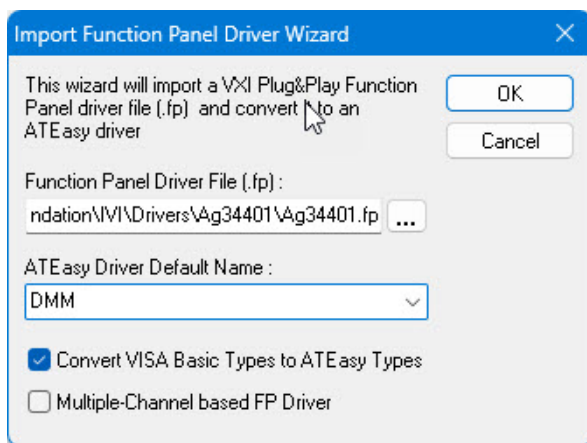


## CYBER SECURITY FEATURES

ATEasy Executable and DLL can be encrypted providing a way to maintain integrity, preventing modifying (non-repudiation), reverse engineering, or PCode viewing. In addition, the ATEasy run-time provides protection against calls stack parameters mismatch and buffer overrun protection after external DLL call. ATEasy source files when saved to binary format can be also encrypted to protect against changes and verify file integrity. Program, System and Drivers files can be password or license protected to limit access to using, viewing or changing the file. Automatic versions records to the source files time and the user making the changes allowing you to track changes beyond using file date.

## INSTRUMENT DRIVERS

In addition to the provided ATEasy instruments drivers, ATEasy provides IVI drivers for all the available IVI classes. ATEasy also supports thousands of VXI Plug-and-Play, Function Panel (.fp files) instrument drivers from vendors such as Agilent Technologies and National Instruments. Using the File Open Command, engineers can open a Function Panel file (.fp) and convert it to an ATEasy driver format (.drv). ATEasy generates a command tree, procedures, DLL function calls, constants and data types as defined in the function panel file and its accompanying C/C++ header file. You can also use LabView drivers since ATEasy can call VI (VI or LLB files). In addition, ATEasy provides various tools to support rapid creation and debugging of instrument drivers.



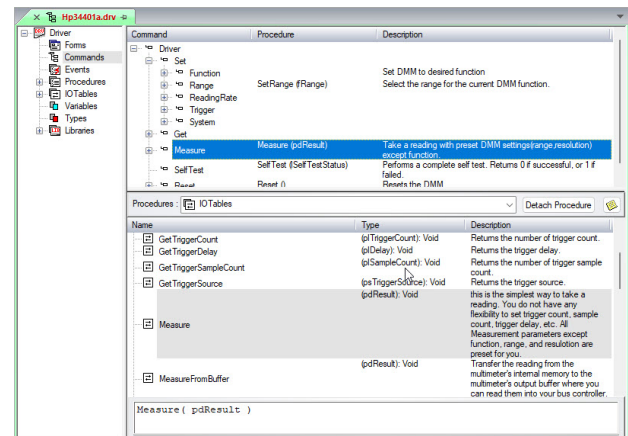
ATEasy Import Function Panel Wizard

## RAPID DRIVER DEVELOPMENT

If none of the above driver formats are available, ATEasy allows engineers to rapidly modify an existing driver or develop new ones. With DLL-based drivers, engineers can easily import existing DLL functions (using C header files or manual methods). .NET or ActiveX/COM based drivers can also be imported and used to create a driver almost instantly. With message-based instrument interfaces such as GPIB, Serial (RS-232), VXI, USB and LXI (TCP/IP). ATEasy uses a simple mechanism called I/O Tables for sending and receiving instrument commands. I/O tables allow ATEasy drivers to be independent of the instrument interface and manufacturer so that one driver can be used with an instrument that supports multiple interfaces (e.g. GPIB and RS232).

## INSTRUMENT INDEPENDENCE

Thanks to ATEasy's exclusive Instrument Interchangeability Technology (I2T), test engineers can easily define plain language driver commands. These commands separate the driver interface used in your test program from the implementation of the driver that uses IO tables, DLLs or any other external library or software component. Within the test program, these commands are independent of the instructions, DLL procedures, and interface type. Consequently, when instruments need to be replaced, a new instrument driver may be used while the test program(s) remains unchanged.



ATEasy Agilent 34401A Driver Commands



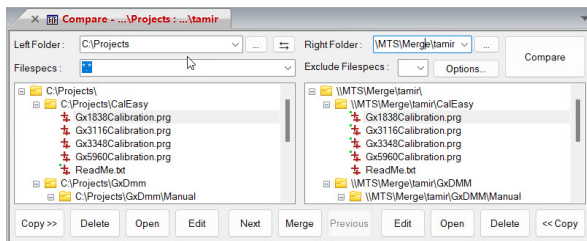
## WHAT'S NEW?

### ATEASY 2023

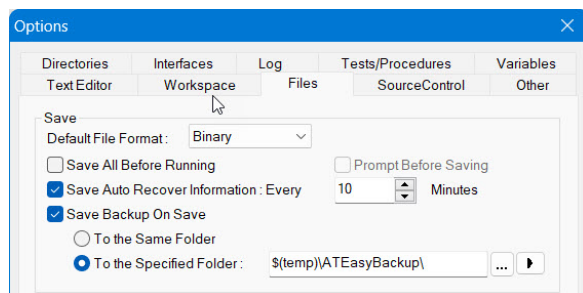
The following list describes the main changes in this version:

#### IDE (Development Environment):

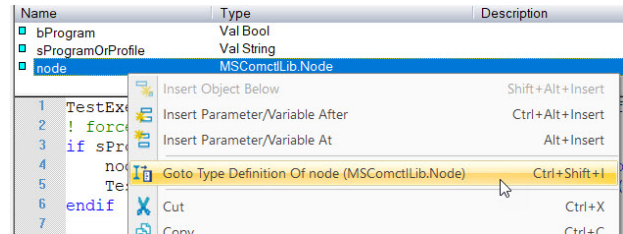
- **Pin tabs** for easier locating you document tab showing the tab always on the same location.
- Reduces tab title (folder path).
- Updated tool bar images and reorganized to show more frequently used button on the left and remove unused buttons.
- **Compare Folders:**
  - Improve speed for large folder comparison
  - Improved file tooltip information.
  - Display green dot next to the newer file.
- After merge/copy/delete is invoked, selection is automatically moved to the next modified/missing file entry in the list.



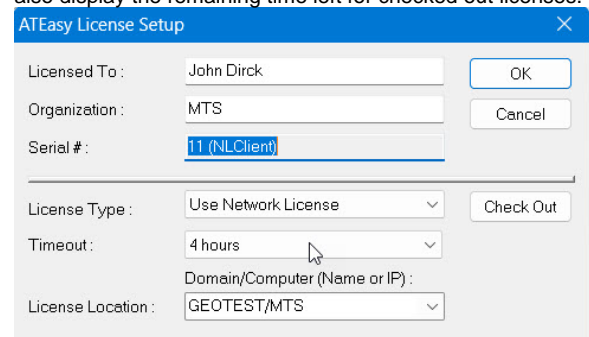
- Options dialog, Files Page:
- **Save Backup On Save** option. This option can be used to save a backup file when saving a file with new changes. Files that are backed up are project, program, system or a driver. Backup files location can be specified to the original folder or any other folder using Path Variables. Files are backed up using the original file name followed by the date and time were the backup created.



- New Code editor/Variables views context menu command, **Go to Type Definition of ...** for Variables and procedures.



- **Licensing**
- Network licensing, extend time out up to 10 days (from 5).
- **Check Out/Check In** buttons in the License Setup dialog, also display the remaining time left for checked out licenses.



#### Run-Time:

- New **aPriorityError** added to enumAThreadPriority used in case GetThreadPriority() receives invalid handle.
- **Peak()** and **Poke()** functions supports 64-bit addressing.
- Physical memory and Port I/O access using internal functions is a privileged instruction and require special access code from MTS.
- New support for inserting .Net control to a from using **AForm.InsertControl method**.
- Improved **Delay()** internal function to support delay of nanoseconds delays instead of milliseconds.
- **#ifndef** statement,
- **Tag** property was added an optional parameter that indicate the ID of the tag allowing storing multiple data tags

#### Test Executive/Profile Drivers

- Redesigned Test Executive tree node storage using the Program, Task or Test objects tag property to improve performance for multi-UUT parallel test
- Redesign Profile driver to support Multiple UUT
- New support to SemiEasy - Semiconductor production user interface add-on
- Tests pane. Add **Duration** to show the time duration it took to execute the selected test



## • System/Program Run/Execution Statistics:

- New Driver parameter "IniWriteStats", if set to 1 program statistics data will be written to the TestExec.ini located in Windows root folder.
- Per Program Information includes:
  - Number of times the program was executed.
  - Number of times the program ended normally (not aborted).
  - Number of times the program ended with PASS status.
  - Total time the program ended with FAIL or ERR status.
  - Time spend running this program (all runs), time specified in seconds.

## • For the System/Test Executive:

- Number of times any program was executed.
- Number of times any program ended normally (not aborted).
- Number of times any program ended with PASS status.
- Total time any program ended with FAIL or ERR status.
- Time spent by the test executive running programs, specified in seconds.
- Time test executive was up and running, specified in seconds.

- New Statistics commands under TestExec Utility IniFile Stats xxx command to get and set program and test executive statistics data to/from the TestExec.ini file. Also included a command to print statistics to the test log (Usually from System.OnEndProgram()).

Stop Time : 12/22/2022 2:50:29 PM  
 Elapsed Time : 4s  
**UUT Status : Pass**  
 Signature : \_\_\_\_\_

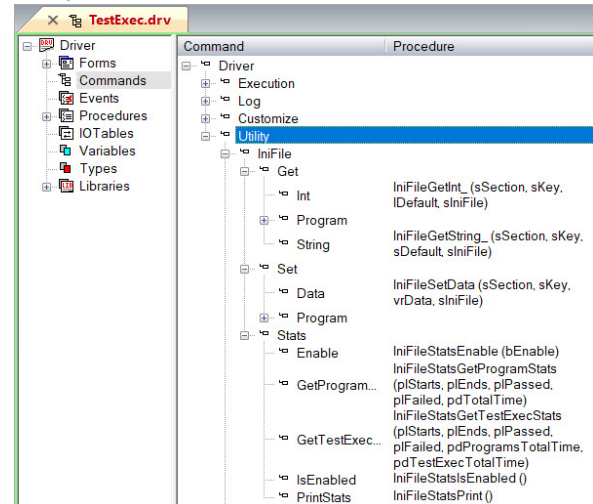
## Program Execution Statistics

Starts: 17  
 Ends: 17  
 Passed/Failed: 17/0  
 Average Program Run Time: 0d:0h:0m:3s

## System/Test Executive Statistics

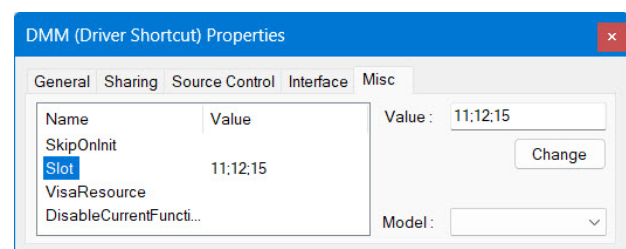
Starts: 17  
 Ends: 17  
 Passed/Failed: 17/0  
 Average Program Run Time: 0d:0h:0m:3s  
 Total Programs Run Time: 0d:0h:1m:1s  
 Total Time: 0d:0h:38m:45s

- New INI File commands under TestExec Utility IniFile xxx command to get and set custom program or system data for INI file.



## MTS Drivers

- Support for multiple cards/domain per one driver shortcut by adding optional last parameters to all relevant commands. This feature allow you to write multi UUT/Site program using App.UutIndex using one driver shortcut. The feature allows you to specify multiple boards or domain during initialization using driver parameters or Initialize command and later on when using the driver command, specify the optional board index to address a specific board. See Driver description for more information.





## ATEASY 2021

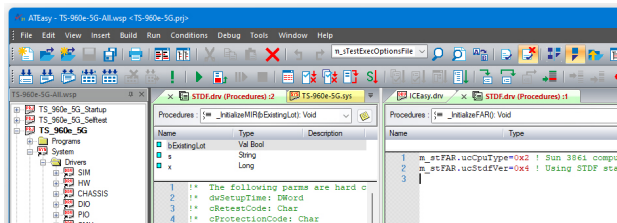
The following list describes the main changes in this version:

- **Revamped IDE user interface** - New tabbed user interface to include improved layout, colored tabs, multi-tab panes, and improved sizing and docking.
- **64-bit support** - additional 64-bit IDE and run-time, ability to mix 32 and 64-bit software components
- **Code Editor/Beautify Code/Auto Completion** - Many, many improvements, including Syntax error highlight, Beautify code, auto completion improvements, improved informational tooltip and parameter suggestion and many more.
- **Compile/Build** - show multiple compiler errors.
- **Cyber Security updates** to enable developers customize the IDE and the test application comply with DOD Application Security and Development (ASD) and DISA Security Technical Implementation Guide (STIG). See [KBase article](#) for more information.
- **~100% backward compatible**, with EXEs, DLLs compiled with older versions, and prior version source files, install side by side with your existing ATEasy version facilitates side by side test application migration

Learn more about ATEasy 2021: [ATEasy 2021 - What's New?](#).

ATEasy 2021 is available for download from the ATEasy **Downloads** tab. See also the ATEasy 2021 preview presentation in the **Tutorials** tab.

The following Image demonstrates the new ATEasy 2021 tabbed user interface:





## SPECIFICATIONS

GENERAL		
<ul style="list-style-type: none"> <li>• Integrated Development Environment for Windows XP-Windows 11 (32/64 bit)</li> <li>• Generate royalty free 32/64-bit Windows executable files (EXE) and DLLs</li> <li>• Provide a framework for test application and for instrument control</li> <li>• Generate Workspace, Project, Program, System, and Driver files</li> <li>• Microsoft TFS, SVN, GIT, Source Safe, and more integration</li> <li>• Files can be saved and load in text or binary format</li> <li>• Built in versioning to each file type</li> <li>• Tools to compare and merge project folders, sources and files</li> <li>• Generate customizable test reports in HTML, MHT or text format</li> <li>• Generate documentation automatically</li> <li>• Multi-level user security</li> <li>• Wizard for generating applications and libraries</li> <li>• Customizable Test Executive with Test Sequencer and Fault Analysis</li> </ul>		
Editing	Windows	
<ul style="list-style-type: none"> <li>• Cut/Copy/Paste/Delete</li> <li>• Multiple Selection</li> <li>• Merge objects</li> <li>• Undo/Redo (unlimited)</li> <li>• Drag/Drop objects</li> <li>• Find/replace for objects name, description, code across multiple modules</li> <li>• Context menu for all objects</li> <li>• Graphical or Text Editor</li> <li>• Tabbed Document windows with Tree/Object views</li> <li>• Workbook with dockable windows user interface model</li> </ul>	<ul style="list-style-type: none"> <li>• Workspace window</li> <li>• Properties window with multi tab (modeless)</li> <li>• Properties Grid window</li> <li>• Variables window</li> <li>• Threads, Calls Stack, Watch and Debug Windows</li> <li>• Build Log</li> <li>• Debug Log</li> <li>• Test and Debug Logs (HTML, MHT or Text)</li> <li>• Find All and Replace All Logs</li> <li>• Source Control Log</li> </ul>	
Code Generation/Editor	Form Editor	
<ul style="list-style-type: none"> <li>• Syntax Highlighting</li> <li>• Font, Tabs, Auto indent</li> <li>• Auto List Members</li> <li>• Auto List Commands</li> <li>• Auto Parameter Information</li> <li>• Auto Parameter Suggestion</li> <li>• Auto Type Information</li> <li>• Batch or Editor Beautify Code</li> <li>• Insert Symbol dialog</li> <li>• Insert Flow control menus</li> <li>• Insert Commands cascading menus</li> <li>• Bookmarks</li> </ul>	<ul style="list-style-type: none"> <li>• Align controls</li> <li>• Same size controls</li> <li>• Space evenly controls</li> <li>• Center across the form</li> <li>• Auto arrange controls</li> <li>• Tab Order controls</li> <li>• Check duplicate mnemonics</li> <li>• Lock Controls</li> <li>• Grid</li> <li>• Margins</li> <li>• Test Form</li> </ul>	
Customization	Options	Help/Documentation
<ul style="list-style-type: none"> <li>• Keyboard keys</li> <li>• Toolbars</li> <li>• Menu</li> <li>• Tools menu</li> <li>• Docked/Float/Tabbed windows</li> </ul>	<ul style="list-style-type: none"> <li>• Directories</li> <li>• Interfaces</li> <li>• Log</li> <li>• Tests</li> <li>• Text Editor</li> <li>• Workspace</li> </ul>	<ul style="list-style-type: none"> <li>• Getting Started manual</li> <li>• User's Guide manual</li> <li>• Reference/Programming Language manual</li> <li>• HTML based help with TOC, Index, Search</li> <li>• Context sensitive help for dialogs and keyword</li> <li>• What's this help</li> </ul>



Compiler	Test Level Debugging	Source Safe Integration
<ul style="list-style-type: none"> <li>• Checkit!</li> <li>• Check Module</li> <li>• Check All</li> <li>• Build</li> <li>• Re-Build</li> <li>• Batch Build</li> <li>• Show Error</li> <li>• Stop Build</li> <li>• Execute</li> </ul>	<ul style="list-style-type: none"> <li>• Loop Task/Test</li> <li>• Skip Task/Test</li> <li>• Current Task/test</li> <li>• Set Next Task/Test</li> <li>• Run to Task/Test</li> <li>• Continuous/Task By Task/Test By Test</li> <li>• Prompt/Pause on Task/Test Failure</li> <li>• Taskit!</li> <li>• Testit!</li> <li>• Log Failures Only</li> <li>• Repeat Run</li> </ul>	<ul style="list-style-type: none"> <li>• Check In/Out</li> <li>• Add Files/Project</li> <li>• Compare Files</li> <li>• Show History</li> <li>• Show Status/Properties</li> </ul>
Source Level Debugging	Debug Windows	
<ul style="list-style-type: none"> <li>• Reset</li> <li>• Abort</li> <li>• Pause/Continue</li> <li>• Step Into</li> <li>• Step Over</li> <li>• Step Out</li> <li>• Breakpoints</li> <li>• Doit!</li> <li>• Loopit!</li> <li>• Formit!</li> <li>• Run to Cursor</li> <li>• Show Next Statement</li> <li>• Set Next Statement</li> </ul>	<ul style="list-style-type: none"> <li>• Calls Stack/Locals with separate tab for module variables and change value</li> <li>• Watch window with expression evaluator</li> <li>• Debug window for executing intermediate code</li> <li>• Monitor (communication) window, with interface source filtering</li> <li>• Threads window for debugging multi-threaded application</li> </ul>	
Interfaces	Other	
<ul style="list-style-type: none"> <li>• GPIB: Keysight, National Instruments and more</li> <li>• VXI: National Instruments - MXI</li> <li>• COMM: Any Windows compatible serial and infrared port</li> <li>• File/Device</li> <li>• LXI, WinSock: Client/Server, UDP/TCP</li> <li>• USB: RAW or USBTMC devices and instruments</li> <li>• ISA, PCI and PXI or any PC-based instruments</li> <li>• None (Other)</li> </ul>	<ul style="list-style-type: none"> <li>• Module events</li> <li>• DLLs</li> <li>• Multi-Threading</li> <li>• Exception Handling (try-catch, error, OnError)</li> <li>• Read C header file for DLL function and data type declaration</li> <li>• Call LabView VIs, Display LLB/VI files (LabView 7.0-8.6)</li> <li>• Import VXI Plug&amp;Play/Function Panel (.fp) drivers</li> <li>• IVI drivers for all available classes</li> <li>• Export VB and C header files for calling ATEasy DLLs from VB or C</li> </ul>	



Statements	Data Types
<ul style="list-style-type: none"> <li>• If - Else</li> <li>• Select - Case</li> <li>• For Next</li> <li>• Repeat - Until</li> <li>• While</li> <li>• Loop</li> <li>• Exitloop</li> <li>• Continue</li> <li>• Run</li> <li>• Exit</li> <li>• Reset</li> <li>• Task</li> <li>• Test</li> <li>• ExitTask</li> <li>• ExitTest</li> <li>• Retry</li> <li>• Ignore</li> <li>• Pause</li> <li>• Abort</li> <li>• Error</li> <li>• Try - Catch</li> <li>• Print/Append/Trace</li> <li>• Add/Remove Handler</li> </ul>	<ul style="list-style-type: none"> <li>• Char</li> <li>• Byte</li> <li>• Short</li> <li>• Bool</li> <li>• Word</li> <li>• WChar</li> <li>• Long</li> <li>• DWord</li> <li>• DLong</li> <li>• DDWord</li> <li>• Float</li> <li>• Double</li> <li>• Currency</li> <li>• DateTime</li> <li>• String</li> <li>• BString</li> <li>• Variant</li> <li>• Procedure</li> <li>• Object</li> <li>• Any</li> <li>• Structure</li> <li>• Enum</li> <li>• Typedef</li> </ul>
Form	COM/.NET
<ul style="list-style-type: none"> <li>• MDI Frame, MDI Child and Normal Forms</li> <li>• Menu Bar and Context Menus</li> <li>• Third Party ActiveX controls</li> <li>• Procedures, Variables, Events</li> <li>• Event programming</li> <li>• Drawing</li> <li>• Tool tip</li> <li>• Win Help/Html Help</li> <li>• Auto size using Anchor/Align</li> </ul>	<ul style="list-style-type: none"> <li>• Load COM/OLE Type Libraries and OCX</li> <li>• Load .NET Assemblies and Controls</li> <li>• Display type libraries/.NET assemblies content</li> <li>• Early and Late Binding</li> <li>• Objects can be use from any thread</li> <li>• CreateObject / GetObject and new operator</li> </ul>



Controls (ActiveX)	UI Classes
<ul style="list-style-type: none"> <li>• AButton</li> <li>• AChart</li> <li>• ACheckBox</li> <li>• AComboBox</li> <li>• ACommonDialog</li> <li>• AGroupBox</li> <li>• AImage</li> <li>• AImageList</li> <li>• ALabel</li> <li>• AListBox</li> <li>• ALog</li> <li>• APanel</li> <li>• ARadioButton</li> <li>• AScrollBar</li> <li>• ASlider</li> <li>• AStatusBar</li> <li>• ATab</li> <li>• ASwitch</li> <li>• ATextBox</li> <li>• ATimer</li> <li>• AToolBar</li> </ul>	<ul style="list-style-type: none"> <li>• AAxis</li> <li>• AControl</li> <li>• AFont</li> <li>• AForm</li> <li>• AImageListImage</li> <li>• AMenu</li> <li>• APicture</li> <li>• APlot</li> <li>• AStatusBarPane</li> <li>• ATabPage</li> <li>• AThumb</li> <li>• AToolBarButton</li> <li>• AVirtualKeyboard</li> </ul>
Classes	Procedure Groups
<ul style="list-style-type: none"> <li>• AApp</li> <li>• AAppFolder</li> <li>• AAppShortcut</li> <li>• AClipboard</li> <li>• ACritical Section</li> <li>• ADriver</li> <li>• AEvent</li> <li>• AMutex</li> <li>• AProgram</li> <li>• ASemaphore</li> <li>• ASystem</li> <li>• ATask</li> <li>• ATest</li> <li>• AUser</li> <li>• AUsers</li> </ul>	<ul style="list-style-type: none"> <li>• ActiveX</li> <li>• Serial Communication</li> <li>• DDE</li> <li>• Driver/Info</li> <li>• File System</li> <li>• GPIB</li> <li>• Interfaces</li> <li>• Interrupts</li> <li>• Log File Information</li> <li>• Math</li> <li>• Misc</li> <li>• Multi-Threading</li> <li>• PC Resources</li> <li>• String Manipulation</li> <li>• Time</li> <li>• USB</li> <li>• Variants</li> <li>• VXI</li> <li>• WinSock</li> </ul>
OS COMPATIBILITY	
ATEASY Version	Windows Client OS
3.0	9x (95/98/Me), NT, 2000
4.0/5.0	9x (95/98/Me), NT, 2000, XP
6.0/7.0	98, Me, 2000, XP, Vista
8.0	98, Me, 2000, XP, Vista, 7
9.0/10.0	XP (sp3), Vista, 7, 8, 8.1, 10 (32/64 bit)
2021-2023/11.0-12.0	XP (sp3), Vista, 7, 8, 8.1, 10, 11 (32/64 bit)

Note: Specifications are subject to change without notice



## ORDERING INFORMATION

GT8000	ATEasy Single User Software Package (includes 1-year subscription & support)
GT8003	ATEasy with a 1-year Subscription and Support and 3-day training at Marvin Test Solutions
GT8320-2Y	ATEasy with a 2-year Subscription and Support
GT8320-3Y	ATEasy with a 3-year Subscription and Support
GT8000-LT	ATEasy Lite, for OEM applications. Supports unlimited OEM drivers and (2) non-OEM drivers, no executable (.EXE/.DLL) generation (includes 1 year subscription and software support). Requires a purchase of GT8000 to start. Note: Only available to OEM system manufacturers for resale of a pre-approved OEM Test System.
<b>SUBSCRIPTION, SUPPORT AND UPGRADE</b>	
GT8020	1-year Subscription and Support for ATEasy (requires continuous agreement)
GT8020-2Y	2-year Subscription and Support for ATEasy (requires continuous agreement)
GT8020-3Y	3-year Subscription and Support for ATEasy (requires continuous agreement)
GT8020-EXP6	1-year Subscription and Support for ATEasy, expired 1 day to 6 months
GT8020-EXP24	1-year Subscription and Support for ATEasy, expired 7 to 24 months
GT8029	1-year Support only for ATEasy (no upgrades)
<b>LICENSING</b>	
ATEasy-XX-YY	ATEasy License Type Replacement for ATEasy Subscribers
<b>TRAINING</b>	
ATEasy-DS	3 days ATEasy training at Marvin Test Solutions (Irvine, CA) for 1-3 persons. Call for larger groups.
ATEasy-DS5	5 days ATEasy training at Marvin Test Solutions (Irvine, CA) for 1-3 persons. Call for larger groups.
ATEasy-DS2	On-site, 3 days ATEasy training seminars for 1-3 persons. Call for larger groups. Note: Price does not include travel expenses. Please contact MTS for a quote
ATEasy-DS25	On-site, 5 days ATEasy QuickStart™ training for 1-3 persons. Note: Price does not include travel expenses. Please contact MTS for a quote
ATEasy-DSW	3 days ATEasy Webinar training for 1-6 persons.
GT98001	ATEasy Training Manual
ATEasy-DSW5	5 days ATEasy Webinar training for 1-6 persons.
GT98901	Multi-Function DAQ, ATE Demo USB Board

GT98901-EDU	ATEasy Intro Kit with a USB ATE Demo Multi-Function Board. Academic use only.
GT98901-INT	ATEasy Intro Kit with a USB ATE Demo Multi-Function Board.
GT98901-UG	ATEasy Intro Kit User Guide
<b>ATEASY 2.X (16 BIT FOR WINDOWS 3.1/9X/NT/2K)</b>	
ATEASY-PC	ATEasy software for PC cards & RS-232 applications
ATEASY-PC-RT	RunTime version of ATEasy-PC (incl Vol I & Getting Started)
ATEASY-GP	ATEasy software for GPIB, PC cards, and RS-232 applications
ATEASY-GP-RT	RunTime version of ATEasy-GP (incl Vol I & Getting Started)
ATEasy-GP-01	ATEasy-GP including GPIB-PCII interface card
ATEasy-GP-02	ATEasy-GP including AT-GPIB Interface card
ATEasy-GP-03	ATEasy-GP including PCI-GPIB-95 Interface card for Windows 95
ATEasy-GP-04	ATEasy-GP including PCI-GPIB-NT Interface card for Windows NT
ATEasy-VX	ATEasy software for VXI, GPIB, PC cards, and RS-232 applications
ATEasy-VX-RT	RunTime version of ATEasy-VX (incl Vol 1 & Getting Started)
<b>GPIB INTERFACE</b>	
PXI-GPIB	PXI IEEE-488 Interface Card PXI Card
PCI-GPIB	PCI GPIB (IEEE-488) Interface Card - PCI-GPIB