EasyStand
TestStand Made Easy

DSE Test Solutions A/S

What Is TestStand?

National Instruments TestStand is the Industry-Standard Test Management Software.

TestStand is a ready-to-run test management software that is designed to help you develop automated test and validation systems faster.

You can use TestStand to develop, execute, and deploy test system software. In addition, you can develop test sequences integrating code modules written in any programming language.

Sequences also specify execution flow, reporting, database logging, and connectivity to other enterprise systems.

TestStand is used by companies that need a scalable, reliable, and efficient solution for test automation in a wide variety of industries, ranging from defense and aerospace applications to medical and life sciences products over renewable energy and everyday consumer electronics.

What Is EasyStand?

EasyStand is a set of tools to make your life with TestStand easy.

Event though TestStand is a comprehensive and flexible tool, it does have shortcomings.

EasyStand offers solutions on issues where TestStand is complex to use or simply does not offer adequate solutions.

EasyStand combines solutions to common technical challenges with a structured approach to project structure and revision control.

This enables you to keep focus on testing your products rather than having to build and maintain expert TestStand knowledge to get your job done.

Use EasyStand to develop and maintain your test systems faster and cheaper gaining on DSE Test Solutions vast experience combined with the industry standard software from National Instruments

TestStand™

NATIONAL INSTRUMENTS™ Silver Alliance Partner

User Interfaces (UI's)

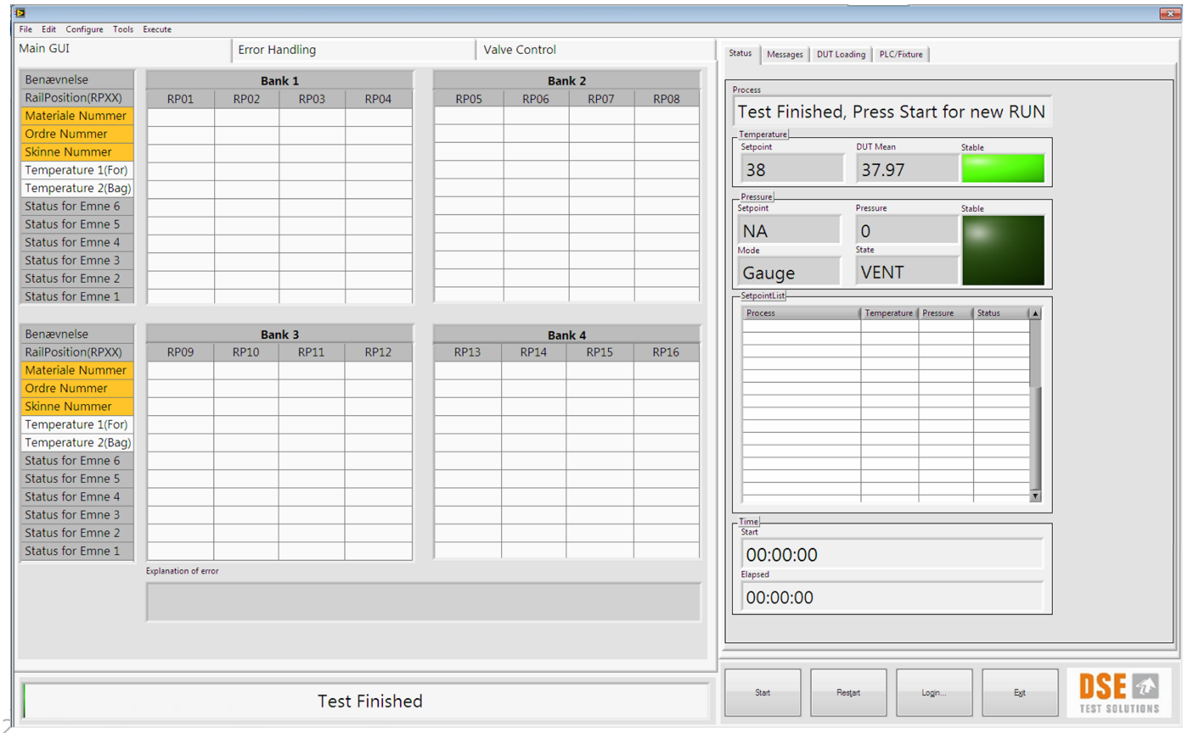
Due to licensing and other issues it is often desired to have a dedicated UI for using TestStand in production environments.
Building modular and scalable UI's is a complex discipline in TestStand. The Standard UI examples shipped with TestStand are **not** good examples on how to do this.

EasyStand UI's are made up of a main UI, working mainly as a whiteboard with tab's and subpanels containing the plugin's and popups that presents the actual UI contents. Using an intuitive configuration editor, test stations can be individually configured to use any combination of main UI, plugin's and popup's desired to fulfil the role of the individual test station, using an intuitive configuration tool.

EasyStand on the contrary enables you to develop and maintain even complex UI's in a very short time.

Key features

- Modular, configurable and scalable UI's, leveraging standard functionality of TestStand combined with LabVIEW project templates for easy design and implementation of Main UI, plugin's and popup's
- Language control of all UI components for easy system deployment in factories around the world offering operators a **UI in the local language**
- Very low coupling between UI and test sequences
- Easy data exchange between UI and TestStand of even large amounts of data
- Popup's can be both floating and embedded in the UI. Popup focus can be automatically given and restored by the system
- Highly scalable with the number of UUT's
- Full support for multi threaded applications
- Standard plugin's and popup's for common TestStand tasks are available
- Object oriented design and implementation



Benævnelse	Bank 1				Bank 2			
RailPosition(RPXX)	RP01	RP02	RP03	RP04	RP05	RP06	RP07	RP08
Materiale Nummer								
Ordre Nummer								
Skinne Nummer								
Temperature 1(For)								
Temperature 2(Bag)								
Status for Emne 6								
Status for Emne 5								
Status for Emne 4								
Status for Emne 3								
Status for Emne 2								
Status for Emne 1								

Benævnelse	Bank 3				Bank 4			
RailPosition(RPXX)	RP09	RP10	RP11	RP12	RP13	RP14	RP15	RP16
Materiale Nummer								
Ordre Nummer								
Skinne Nummer								
Temperature 1(For)								
Temperature 2(Bag)								
Status for Emne 6								
Status for Emne 5								
Status for Emne 4								
Status for Emne 3								
Status for Emne 2								
Status for Emne 1								

Explanation of error

Test Finished

Process: Test Finished, Press Start for new RUN

Temperature
Setpoint: 38
DUT Mean: 37.97
Stable: 🟩

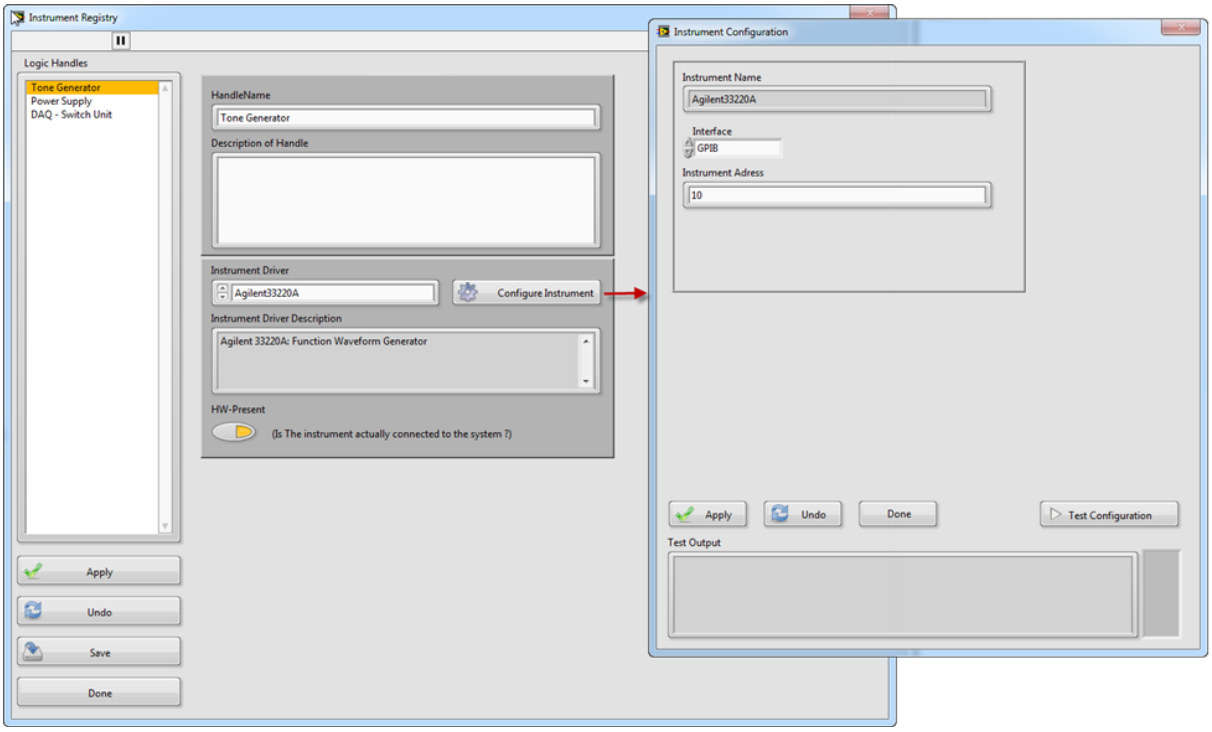
Pressure
Setpoint: NA
Pressure: 0
Mode: Gauge
State: VENT
Stable

SetpointList

Process	Temperature	Pressure	Status

Time
Start: 00:00:00
Elapsed: 00:00:00

Instrument Drivers



Instrument Registry

Logic Handles: Tone Generator, Power Supply, DAQ - Switch Unit

HandleName: Tone Generator

Description of Handle:

Instrument Driver: Agilent33220A

Instrument Driver Description: Agilent 33220A: Function Waveform Generator

HW-Present (Is The instrument actually connected to the system ?)

Instrument Configuration

Instrument Name: Agilent33220A

Interface: GPIB

Instrument Adress: 10

Test Output:

EasyStand leverages a combination of an object oriented approach and LabVIEW project templates, to make it easy to create new instrument drivers either from scratch or by inheriting from existing drivers.

Configuration editors etc. are already present and generalized so more or less all that is needed, is to write the code that actually communicates with the instrument

The concept offers very simple and logic use of instruments in your test sequences.

Key Features

- Auto registration of all instrument drivers present on the system
- Instruments are referenced using logic handles. This means instruments can change address etc. without affecting the test sequences
- General Instrument configuration UI handles configuration of all existing and future instruments
- Very open design. Put in any features of any instrument it can still be handled by the concept
- Easily test that the instrument is connected already at configuration time rather than run time
- Simulate all or individual instruments, enabling you to write and test your test sequences before you have instrument hardware available
- Create instrument families with interchangeable instruments so that you can switch instrument manufacturer without changing a single line in the test sequences
- Easily handle multiple instruments of the same type using logic handles
- Instrument configuration stored in XML format either locally or in database
- Very little coding for new instrument drivers: 3 simple VI’s and you are up and running with a basic instrument

Diversity/Variant Handling

It is often a fact that products comes in many variants and revisions. Combined with the fact that it may undergo several production processes and your test sequences may be more or less completed at any given time during the development process, a comprehensive diversity tool is needed.

Diversity Handling lets you define your products, production processes, versions and release states. You can even add your own categories of diversity to define your production setup.

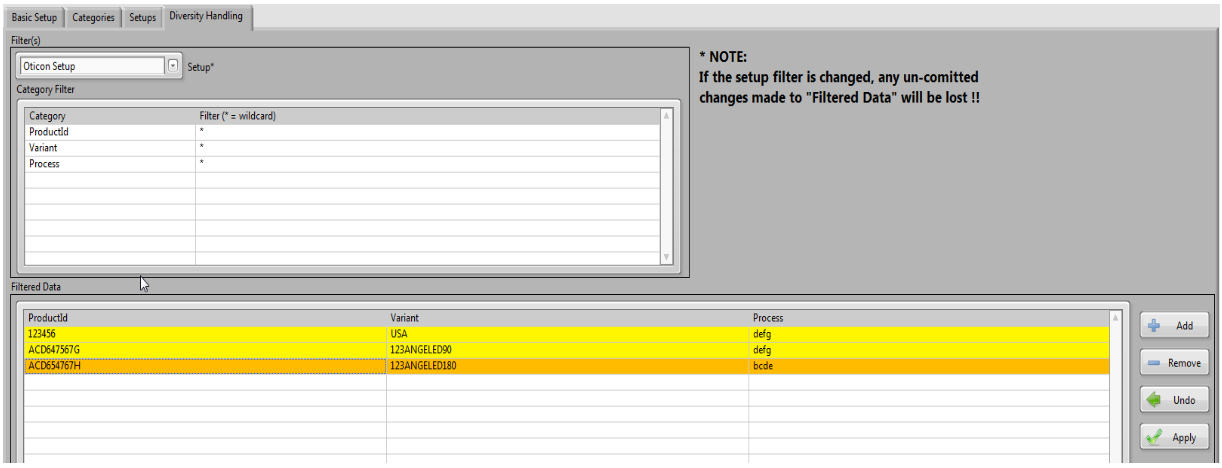
When TestStand executes your test, Diversity Handling will ensure the use of the correct test sequence, limits, firmware file etc. for the UUT, based on the current situation.
Diversity Handling is open to be combined with identifying your UUT's, processes etc. using barcodes, data exchange with existing enterprise systems or any other way available.

Diversity Handling basically works by exporting the parameters (properties) of interest to a database. The properties can be edited outside the TestStand sequence editor in the “Property Value Editor” under full revision control. At the time of testing the values are imported back into TestStand and used during test.

Without Diversity Handling you will probably end up with many sequences that are almost identical but with slightly variant depending contents. Maintenance of these many sequences are tedious and error prone and over time the main reason for inconsistency in your product test.

Key Features

- Control the value of any property of interest in a test sequence
- Intuitive Editor for editing the values of all properties under version control
- Very simple selection from the TestStand sequence editor of the vast number of properties that can be controlled
- Simple import of properties directly from the database or from local data, synchronized with the database enabling offline production
- Revision control ensures that no property can be deleted as long as the product is maintained in diversity control. You can always recreate the conditions under which a specific UUT was tested
- Diversity handling is based on a proven standard concept of handling diversity using TestStand properties, a native part of TestStand



Setup*: Oticon Setup

Category Filter

Category	Filter (* = wildcard)
ProductId	*
Variant	*
Process	*

*** NOTE:**
If the setup filter is changed, any un-comitted changes made to "Filtered Data" will be lost !!

Filtered Data

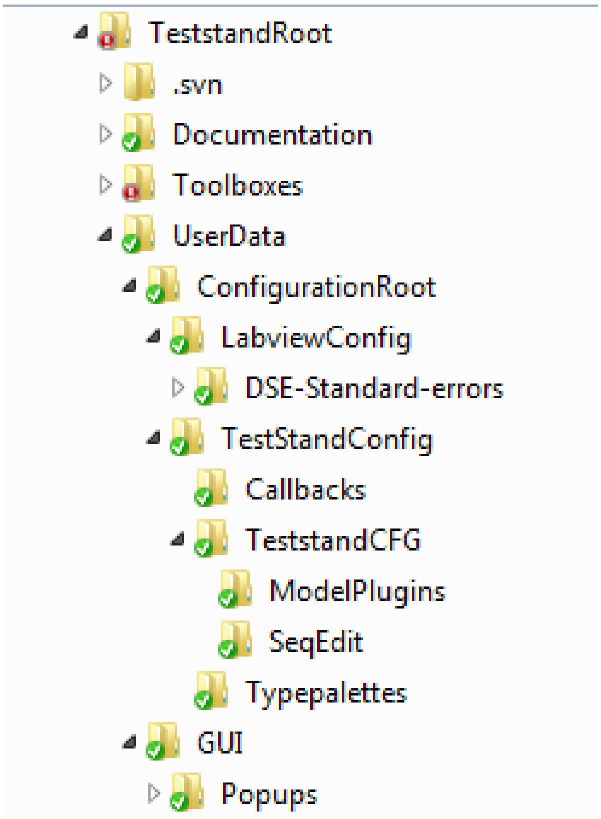
ProductId	Variant	Process
123456	USA	defg
ACD647567G	123ANGELED90	defg
ACD654767H	123ANGELED180	bcde

Flexible project structure

All projects based EasyStand leverage a common very flexible project and file system structure. All references to files are made via handles. No direct references to the file system. This means the project structure is fully re-locatable and re-definable via a tool called the TagEngine combined with clever dynamic linking.

Key Features

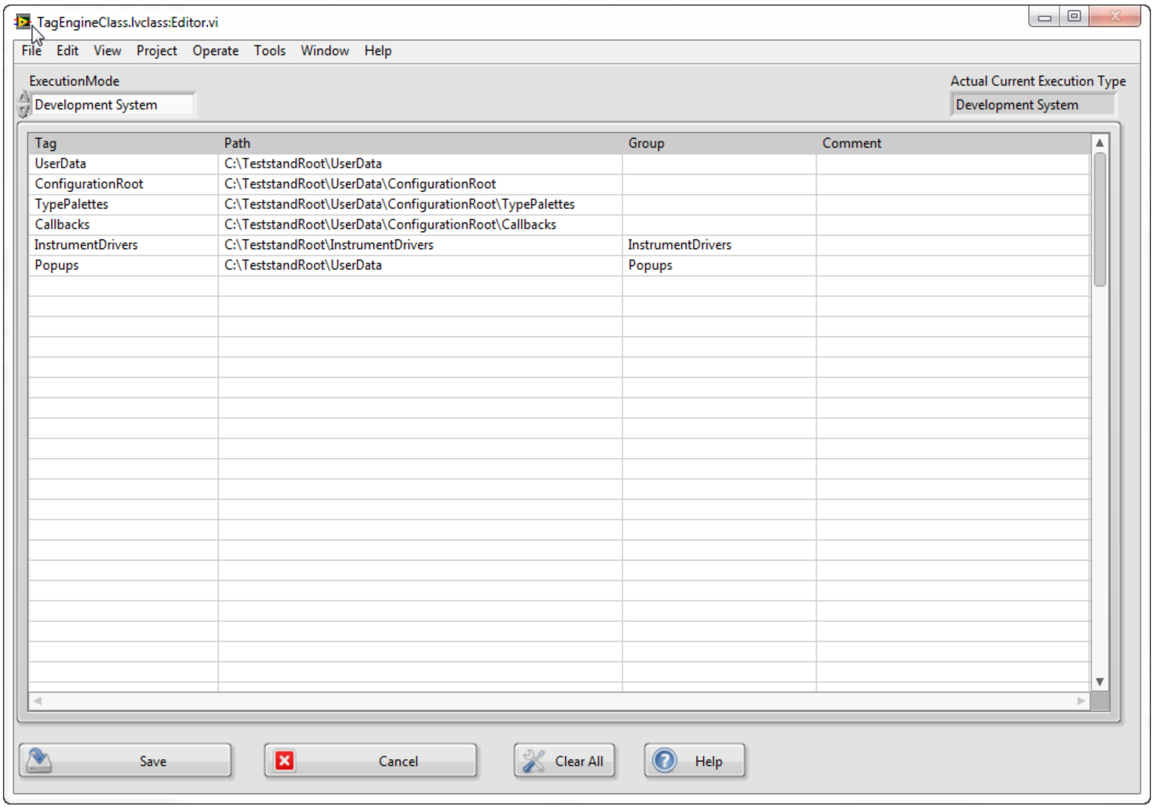
- Dynamic Linking makes it possible to keep ALL files in the project, including TestStand and LabVIEW configuration files under full Source Code Control (SCC)
- High degree of freedom to keep a logical and flexible system design even as the test environment requirements develops over time during the lifespan of your test systems
- Contributes to easy test station configuration and deployment using standard SCC tools
- Logical project file system structure to ensure project overview



```
TeststandRoot
  .svn
  Documentation
  Toolboxes
  UserData
    ConfigurationRoot
      LabviewConfig
        DSE-Standard-errors
      TestStandConfig
        Callbacks
        TeststandCFG
          ModelPlugins
          SeqEdit
        Typepalettes
    GUI
      Popups
```

Example of Project structure

The above example shows an example of a project structure that includes all LabVIEW and TestStand configuration files within the project structure.



TagEngineClass.lvclass:Editor.vi
File Edit View Project Operate Tools Window Help

ExecutionMode: Development System

Actual Current Execution Type: Development System

Tag	Path	Group	Comment
UserData	C:\TeststandRoot\UserData		
ConfigurationRoot	C:\TeststandRoot\UserData\ConfigurationRoot		
TypePalettes	C:\TeststandRoot\UserData\ConfigurationRoot\TypePalettes		
Callbacks	C:\TeststandRoot\UserData\ConfigurationRoot\Callbacks		
InstrumentDrivers	C:\TeststandRoot\InstrumentDrivers	InstrumentDrivers	
Popups	C:\TeststandRoot\UserData	Popups	

DSE Test Solutions A/S
Sverigesvej 19
DK-8700 Horsens
Denmark

Tel: +45 75618811
Fax: +45 75615895
Web: www.dse.dk
Mail: dse@dse.dk

QUALITY SYSTEM CERTIFICATION
DNV·GL
ISO 9001
DNV GL BUSINESS ASSURANCE DENMARK A/S

The history of DSE Test Solutions A/S

The history of DSE Test Solutions A/S goes back to 1981, and for more than 25 years we have developed and manufactured test systems for various types of production tests and verification tests. Furthermore we also produces various types of wire testers and moisture meters for straw and other residues from the agricultural sector.

The vision of DSE Test Solutions A/S

It is our vision to be the customers preferred supplier of test solutions and products for securing quality.
To secure this:

- **We always place the customer in center**
- **We lead decentralized with focus on the competences of our employees with very few rules to support the development process in order to avoid mistakes**
- **We aim to be a well consolidated company with the necessary means to develop the company during periods with high as well as low growth.**
- **We want to be known as one of the most attractive companies in our region, in order to attract - hold on to - and develop the best employees in the business.**