Demo Script for Real-Time Test Automation (RTTA)

*Contact Ricardo Santa Olalla or Chris Washington with any questions*

This demo goes along with the slides called RTT Automation.pptx.

The purpose of this demo is to show the different test components and software tools used in a Real-Time Test, how to automate them and how to integrate them in a single tool. The demo is broken into three main sections that go through configuring everything manually, then explaining an automated test sequence and finally how to use the RTTAutomation Reference Design to abstract and automate these software tools.

# Sections

### [Manual Demo](#_Manual_Demo)

The Manual Demo will guide you through create every test component you need from scratch. This means creating a Req Gateway project, a VeriStand project and a DIAdem script.

[Requirements Gateway](#_Requirements_Gateway)

[VeriStand](#_VeriStand)

[DIAdem](#_DIAdem)

### [Automated Demo](#_AUTOMATED_DEMO)

The Automated Demo Section is for you to explain the important parts of a TestStand sequence that automates VeriStand and DIAdem to perform a Real-Time Test.

### [Scripting Demo (aka NIGEL) using RTTAutomation](#_SCRIPTING_DEMO_(aka)

The NIGEL demo section will explain how to use the RTT Automation Reference Design to demo NIGEL.

[MIL](#_MIL)

[MIL to HIL](#_MIL_to_HIL)

# Manual Demo

***NOTE****: The following steps guide you through creating the different test components from scratch. If you don’t have enough time for this, you can use the already created and configured components that ship with the NIGEL demo and use them to explain the concepts.*

## Requirements Gateway

***NOTE:*** *Before running Requirements Gateway, make sure you don’t have any TestStand Sequence or TestStand report in the demo’s Working Directory from previous demo runs. Otherwise you will see the incorrect number of covered and uncovered requirements.*

1. Open Requirements Gateway and open create a new project.
2. Go to File >> Edit project
3. In the Configuration Window add a new document by clicking in the open book and placing the box inside the white are.
   1. In the detail window, click under type of analysis and select Word
   2. Click then under File or Directory and browse for the Engine Test Requirements document.
   3. Click OK and then Yes when prompted to reanalyze the project

As you will see the document has been added to the req gateway project but the document itself has three requirements gathered from the document. The document should have a total of four requirements. To make sure that req gateway finds all the existing requirements you need to edit the document and specify what a requirement is by defining the style used in that line. The following steps will guide you through that

1. Double click in the word document icon inside the Req Gateway project.
2. Look for section 3.2.1 Coastal Summer-Engine Speed Test.
3. Highlight the REQ\_CoastalSummer\_Condition line and change its style to Requirement \_ID
4. Hightlight the following 3 lines and change their style to Requirement text
5. Save the document. Go back to Req Gateway and click yes when prompted to reanalyze the project.

Now you should have 4 requirements. The next thing is to add the covering document or TestStand sequence that will cover the requirements defined in the document.

Here you have two options. If you want to do the RTTA demo then I recommend you to configure Req Gateway completely but save more explanation for later. The following steps will guide you through the configuration. If you want to fully demo Req Gateway at this point then you will have to do some modifications to the following steps. I’ll mark those modifications with an \*.

In the case of only configuring Req Gateway, since we don’t have a TestStand sequence yet we will only add a path to where the sequence will exist.

1. Go back to edit the project
2. Add another document just underneath the existing word document.
3. Modify its type to be a TestStand Sequence
4. Then click again under path and point to the Working directory.

\*\* 12\*\*) If you want to demo it now, then instead of adding a directory you will add the TestStand sequence that covers the requirements.

NOTE: To generate a TS sequence and report to be added in this step I recommend you to run NIGEL/RTTA once yourself so that you can obtain the files.

1. Once that configured click on add a cover and click from the TestStand sequence block to the word document block. This will create a “wire” that represents coverage.
2. Click Ok and Yes when prompted.

This will make your 4 requirements go red under the project and on the right it will say that you have 4 uncovered requirements.

Since we want to not only know that there is a sequence that covers the requirements but also know the results of the test, we need to add a TestStand report as a covering document for the TestStand Sequence. Since we don’t have a Sequence or report yet, we add an empty path again. The following steps will guide you through adding a TS report to the project.

1. Go back to Edit project and add another document right underneath the TestStand document.
2. Edit its type to TestStand report.
3. Edit its path to the same working directory where the sequence will be saved.

\*\* 17\*\*) If you want to demo it now, then instead of adding a directory you will add the teststand report that covers the requirements.

1. Click on add coverage and wire from the Report box to the Sequence box.
2. Click OK and Yes when prompted.

Skip to # bla if you want to fully demo Req Gateway, otherwise continue with the next steps.

## VeriStand

With this, we have configured requirements Gateway. There is no need to explain anything more at this point. The next step is to create the VeriStand project from scratch.

1. Open VeriStand, create a new VS project and save it wherever you want. You might want to use the default VS path or use the folder that I use in the demo to store VS projects
2. Click on Configure project if project didn’t open.
3. Go to System Defintion Files and double click on the nivssdf project
4. Since we are going to do MIL first then click on the Controller and make sure the Operating System is Windows and change the Target Rate to 200Hz
5. Add the following models making sure to import all their parameters and signals:
   1. Application\_mdl.dll
   2. Controller\_mdl.dll
   3. Engine\_mdl.dll

You can find these model DLLs in the model folder for RTTA.

1. Click on System Configuration Mappings and bind:
   1. Controller / Outport/ Fueling -> Engine/Inport /Fueling
   2. Engine /Outport/IMP -> Controller/Inport/IMP
   3. Engine/Outport/IMT - > Controller /Inport/IMT
   4. Application /Outport/Load Torque -> Engine/Inport/Tload
   5. Engine/Outport/RPM -> Controller/Inport/RPM
   6. Engine/Outport/RPM -> Application/Inport/RPM
   7. Application/Outports/Throttle -> Controller/Inport/Throttle
   8. Controller/Outports/Timing -> Engine/Inport/Timing
2. Save the config and close
3. Click on Operate and deploy
4. Open the workspace .nivsscreen and add the following items
   1. Odometer and map it to Engine/Outports/RPM
   2. An auto temp gauge and map it to Engine/Outports/IMT
   3. An auto pressure gauge and map it to Engine/Outports/IMP
   4. A numeric control and map it to Application/Inports/Speed Profile

This control allows the user to modify the setpoint manually (try 1000 or 1500)

* 1. A Graph/rpm torque graph and map it to :Engine/Outport/RPM and to Application/Outport/Load Torque

This Graph will display the rpm vs torque and compare it to a mask.

* 1. A graph and map it to : Application/Inports/RPM and to Engine/Outport/RPM.

This graph will display the speed setpoint driven by the user or by the stimulus profile vs the engine output.

* 1. A model calibration list and map to Engine/Tamb(degF)/Value and Engine\_mdl/Pamb(psi)/Value

NOTE: You will need to first deploy the configuration for you to see these parameters as an option.

1. Play changing the pressure and temperature and the setpoint. Explain how the model is already running and how you use VeriStand UI to modify the parameters and access the model outputs.

Next step is to show how to use the stimulus profile.

1. Close the workspace and click on Profiles/ Add / New Stimulus Profile. You might want to add an existing one if you want to reuse the one I already configured, if not:
2. Go to the profile tab and add change the Generator Name to Speed Profile
3. Click on Add Mappings and map it to Application\_mdl/Inports/Speed Profile
4. Click on Add and add a Replay step
5. On the right side select the data tab and click inport data and browse to the engine\_speed1000.txt located in the Stimulus Profiles/FTP Driving Cycles folder
6. Click OK and save the Stimulus Profile

This stimulus profile will be used as a setpoint for driving the engine’s rpms up and down.

1. Go to the Logging tab and click on the + button at the button.
   1. In the File Path just enter Engine Test.tdms
   2. Then click on the Channels tab and add the following channels:
      1. App/Inports/SpeedProfile
      2. Engine/Outports/RPM
      3. Engine/Param/Pamb(psi)/Value
      4. Engine/Param/Tamb(degF)/Value
      5. Engine/Signals/Tload: 1
      6. Application/Outports/Load Torque
2. Save the stimulus profile file.
3. Close the Stimulus Profile Editor and save the project.
4. Go to operate and select run. If it is grayed out, just double click on the workspace.

You will see all graphs updating. Try again modifying the speed profile to 1500

1. Without closing the Workspace go back to the project and doubleclick on the stimulus profile.nivtest file. Then click on run.

While running if you go back to the workspace you will see the second graph showing the rpms and the setpoint. The first graph will take a while to update.

Once it is done you can close the profile editor.

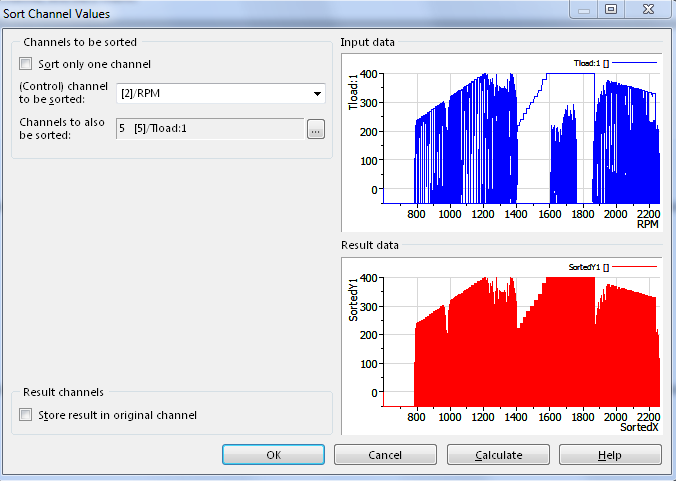
You might want to explain at this point that to perform the test required in the requirements document you would need to first set the temperature and pressure to the values specified in the document and then run the stimulus profile.

## DIAdem

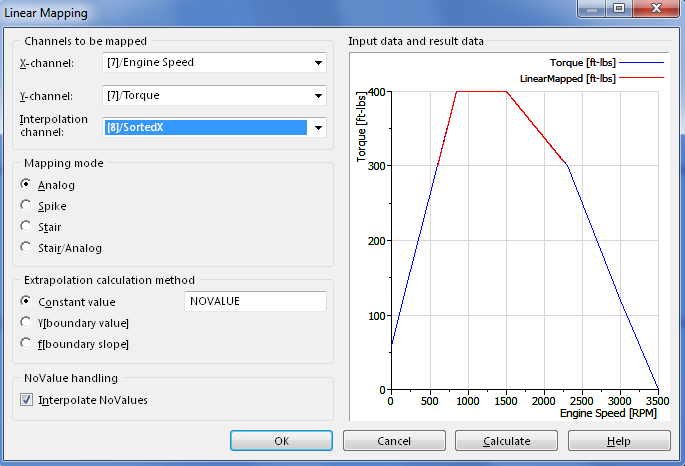
At this point you have configured and ran VeriStand. Unfortunately there is no direct path to link VS to requirements gateway so all analysis would have to be done manually. The next step (skipping TestStand for the moment) is to do the analysis using DIAdem.

***NOTE:*** *In the Scripting Demo section it is recommended to use the shipping DIAdem script for the NIGEL demo. If you want to create your own, Open DIAdem and enable Macro recording under the scripting section and follow the steps below.*

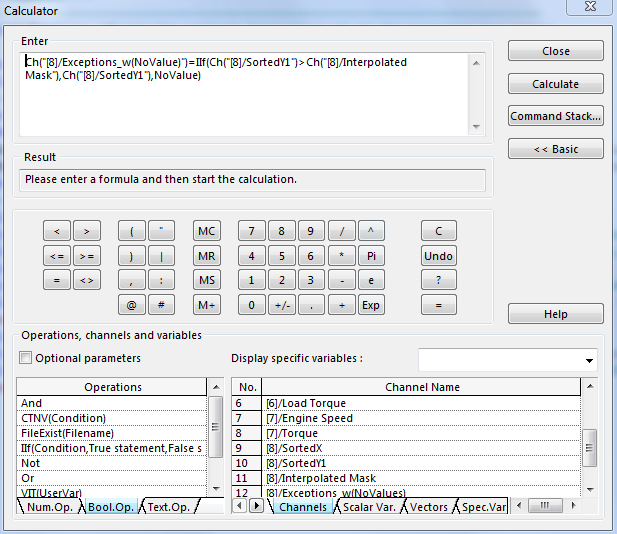
1. Open DIAdem
2. Load Engine test tdms file newly created after running the Stimulus Profile once.
3. Load Mask TDM (from : Demos\EngineController\Analysis Scripts\Mask\_A Data.TDM)
4. Create Group (Sorted Data)
5. Analysis Palette:
   1. Sort Channel Values
      1. First Channel = Engine Output> RPM
      2. Second Channel = Engine Signals > Tload:1
      3. Store results in new channels in the Sorted Data group: (By default SortedX and SortedY)



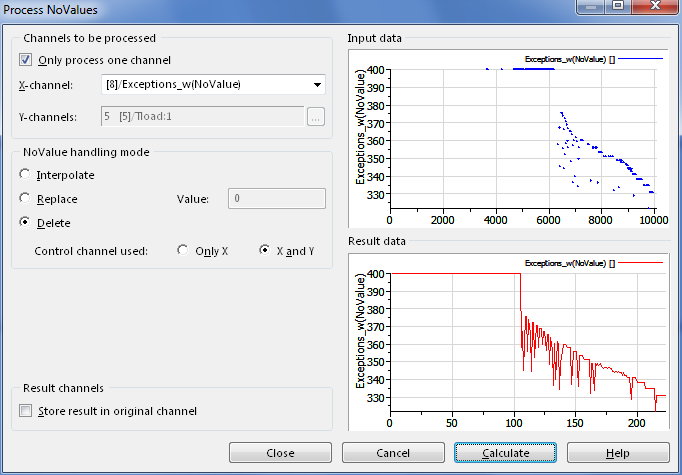
* 1. Interpolate Mask : Linear Mapping
     1. X-Channel = Mask Data/ Engine Speed
     2. Y- Channel = Mask Data/Torque
     3. Interpolation channel = Sorted Data/Sorted X
     4. Store results in a new channel and rename it as Interpolated Mask



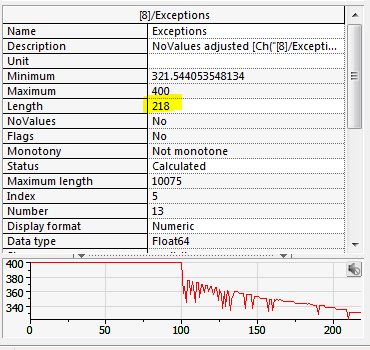
* 1. Create a new channel called “Exceptions\_w(NoValue)”
  2. Use the Calculator to find out the exceptions
     1. Ch("[8]/Exceptions\_w(NoValue)")=IIf(Ch("[8]/SortedY1")>Ch("[8]/Interpolated Mask"),Ch("[8]/SortedY1"),NoValue)



* 1. Delete No values – Call Process NoValues
     1. Check the Only process one channel check box
     2. X-Channel = Excepetions\_w(NoValues)
     3. NoValue handling mode = Delete
     4. Control channel used = X and Y



* + 1. Rename channel to Exceptions
  1. The number of exceptions is the length of the last channel created (Exceptions)



1. The next step is to create a report, so you can go from the VIEW and generate the report from it.

# AUTOMATED DEMO

NOTE: To do this demo you will need a TS sequence. Before presenting you should use the RTTA ref design with the NIGEL demo to generate a TS sequence.

At this point you have demonstrated how to use Req Gateway (partially), VeriStand and DIAdem. The next natural step is to talk about how to automate everything you have done. The answer to that is using TestStand. It depends how much you want to talk about TS but the key parts is that TestStand is used to call the different software tools (VS and DIAdem) through their APIs (.NET and ActiveX) and tell them what to do and when. This means that before automating anything, you first had to configure VS and DIAdem just like you did. You might also want to bring up the point that TestStand does have a way to link back to requirements gateway and that is why you added a folder where a TS sequence will exist to cover the requirements.

# SCRIPTING DEMO (aka NIGEL)

To Demo NIGEL using the RTTA Automation use the already configured VS, Req Gateway, and DIAdem files that ship with the demo, unless you have enough time to test/troubleshoot the files you created in the Manual Demo section.

***NOTE:*** *Before you run NIGEL make sure you don’t have any TestStand Sequence, TestStand Report, or DIAdem reports in the demo’s Working Directory folder. Otherwise you might see the incorrect number of uncovered and covered requirements in Req Gateway.*

***NOTE:*** *The demo ships with an already configured script called the Engine Test Script.rtts. If you don’t have time to build a demo from scratch using the following steps you can open this script and use it to explain its components, generate the TS sequence and run the test. Go to step 22 for going from MIL to HIL.*

### MIL

1. Open Requirements Gateway
2. Open the shipping project called NIGEL Dynamic Test Req.rqtf
   1. Review with the audience how the Word document that contains the requirements has been linked to the project and how the project is also expecting a TestStand Sequence (For more info on this review the Manual Demo > Requirements Gateway section).
   2. Explain how you have 4 uncovered requirements.

**NOTE**: *If you have more a number different from 4 as uncovered requirements, go to the Working Directory folder and delete any TestStand sequence and any XML report.*

* 1. Explain that this is all you will cover right now on Req Gateway but you will get back to it.

1. Open RTTA
2. Go to the Test Script Builder and click on New
3. Give it a name (Engine Test Script1) and save it in your working directory
4. Select yes when prompted to link the script to a req doc. Browse for the req doc.
5. Right click on the Project file folder and add the MIL\_EngineController.nivsproj and click OK
6. Right click on the Stimulus Profile folder and select add a Stimulus Profile. Add the FTP1000.nivstest file and click OK.
7. Right click on the Analysis Script and select to add a script. Add the MaskA\_Analysis.VBS file and click OK.
8. Right click on the MIL\_EngineController.nivsproj and add a calibration file.
   1. Select the Coastal Summer conditions and click OK.
9. Rightclick on the CalibFile\_CoastalSummer.csv calibration file and add a req. Link it to the Coastal Summer conditions req and click OK.
10. Rightclick on the FTP1000 stimulus profile file and add a req. Link it to the FTP1000 req and click OK.
11. Rightclick on the MaskA\_Analysis.VBS analysis script file and add a req. Link it to MaskA req and click OK.
12. Save the script
13. Go to test sequence builder and generate the sequence
14. Go back to Req Gateway and show that all except one requirement are black. This means that there is a test that will cover those requirements. Explain that you now have more uncovered requirements because now Req Gateway is expecting a TS report.
15. Go back to the RTTA and click on Single Pass.
16. Once finished go back to Req Gateway and go to Impact Analysis.
17. In the middle column browse to the different req and notice how on the right hand side you see the TS step that covers that req.
18. Browse to the analysis requirement and notice how on the right side not only you see the TS step but you also see that it failed.
19. Next step is to go back to RTTA to the report viewer and open the TS and the DIAdem reports.

At this point go back to the slides and do a quick summary before talking about how to go from MIL to HIL.

### MIL to HIL

1. To go from MIL to HIL in the script tree click on the test procedure (MiL Test) and while holding your control key drag the Procedure to the Engine Test Script1.rtts title or tree root.

This will make a copy of the Test Procedure.

1. Right click on the second MiL Test section, right click on it and choose rename. Name it HiL Test.
2. In this new HiL section, right click on the MIL\_EngineController.nivsproj and select replace project file.
3. From the drop down list, change MIL to HIL.
4. Select the HIL\_EngineController.nivsproj and click OK.

This should replace the MIL proj file with the HIL file.

1. Save the script.
2. Explain that now you have a test script that will perform MIL and HIL using the same test components (stimulus profile, analysis and parameters).
3. Go to the Test Sequence Builder and generate the TestStand sequence.
4. Stop at this point

***Note:*** *To do a real HIL system you could install VS on a cRIO and run the controller model in it. You will then have to modify the VS HIL\_EngineController system configuration file to point to your PXI system, remove the controller model and map the model inports and outports to hardware.*