**Export Test**

**Coding Concept**

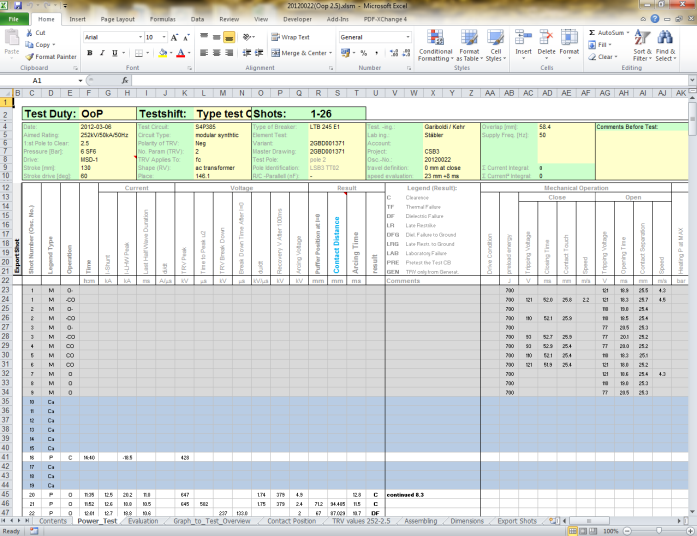
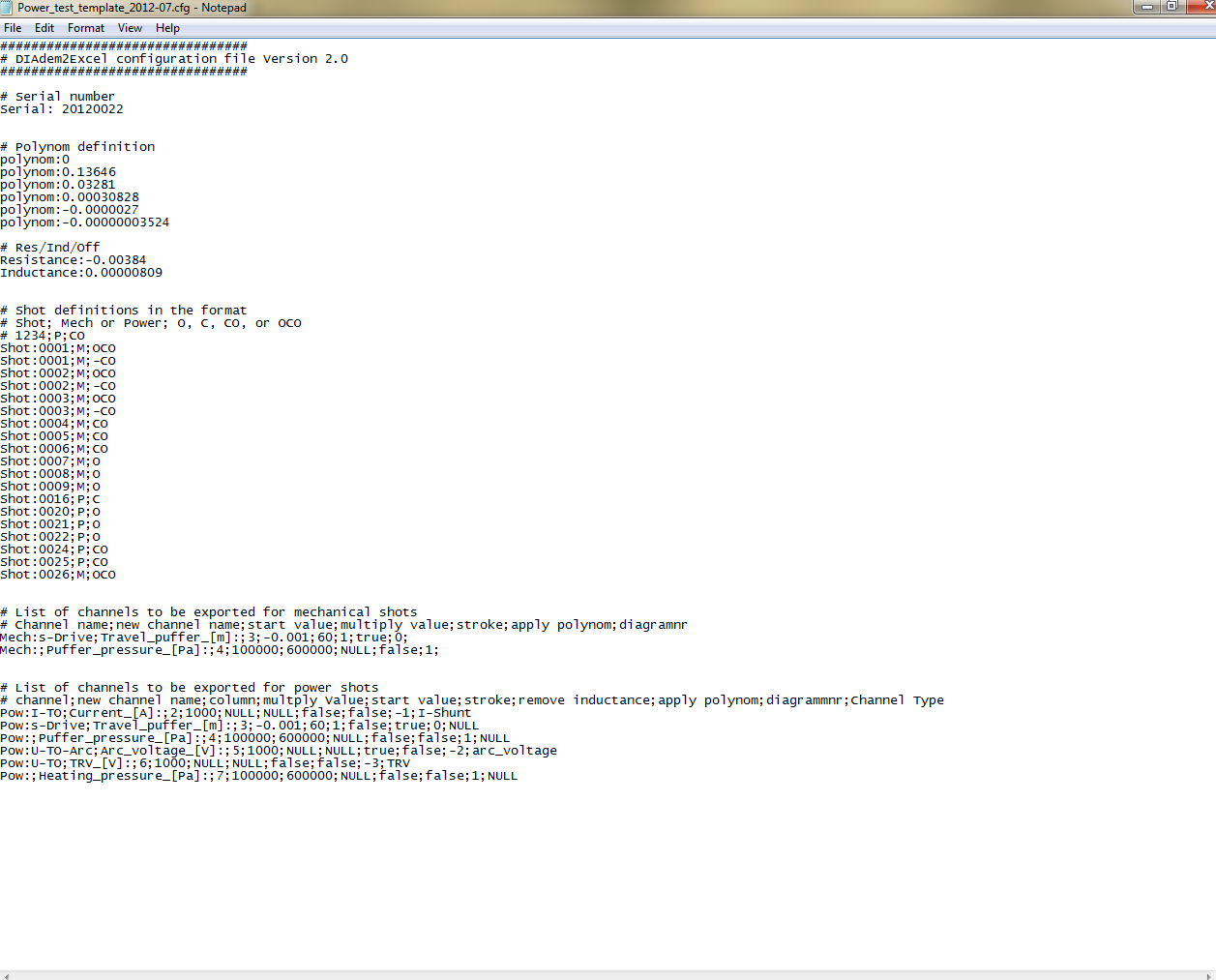
Version 2.0 Written 09.01.2013

**Overview**

The purpose of the files in the Export Test folder is to extract, organize and filter test data from DIAdem Database for different analysis.

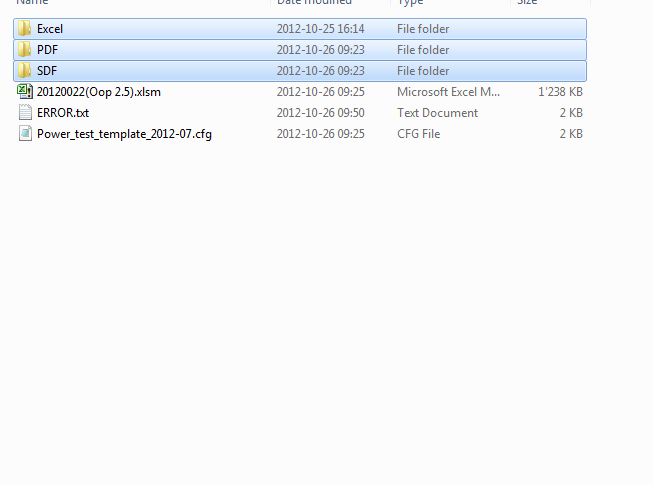
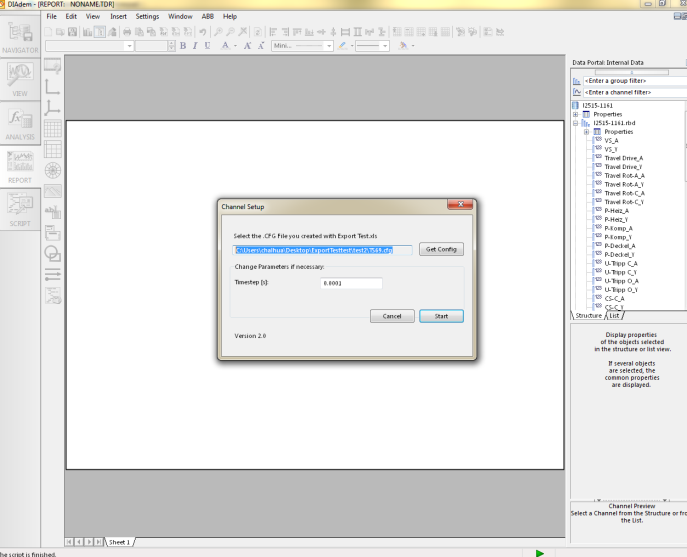
Two steps are required to complete this task. First one needs a test report filled out by test engineers as input. Using the Excel sheet “Export Test”, a configuration file is created.

Input Configuaration File

Second the configuration file is used as input to DIAdem and three separate folders (Excel, PDF, and SDF) are created. These folders contains the filtered and organized the data.

Configuration File Input Output Folders



This document aims to explain the coding behind the second step; as it can be complicated at the first glance. For more instruction regarding how to use the Export Test program, please read “Export Test Instruction for Use”.

**Code Explanation**

Here are the required files for Export Test to function.

ClearTravelREPORT.TDR

ClearTravelSpikes.vbs

ExportTest.vbs

ExportTest\_functions.vbs

ExportTestWind.sud

ExampleREPORT.TDR

MechREPORT.TDR

Output\_Functions.vbs

PowerREPORT.TDR

Report.TDR

WindowsZip.VBS

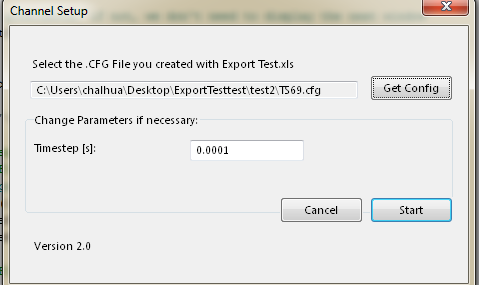
Most of the functions are written in the one of the two files(ExportTest.vbs and ExportTest\_functions.vbs). This file describes some of the important and difficult functions as to purpose and implementation.

*ExportTest.vbs*

sub CallDialog():

This function initialises the export process.

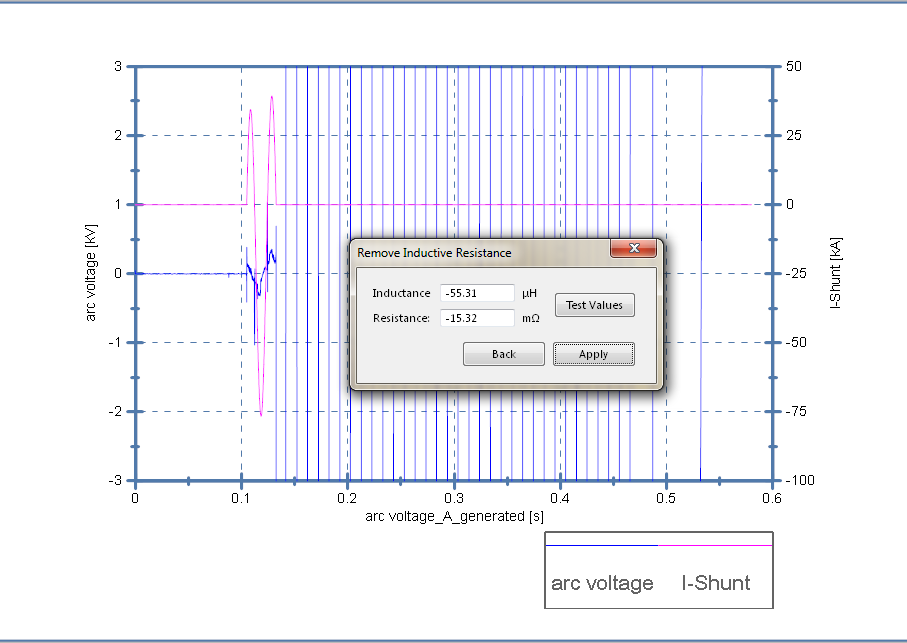
* Uses Suddlgshow("ChannelSetup", myFolders(3)) function to open up the ExportTestWind.sub window for user to load the configuration file and set Time Step.



* calls LoadCfg(cfgFile) to read in and parse the configuration file created from excel in order to understand the parameters of the export. e.g. shot numbers, channels, etc.
* If the configuration file indicates at least one power shot is to be exported, arc voltage filtering process is started by calling LoadExampleShot(firstPowerShot,SerialNumber)

LoadExampleReport(firstPowerShot,SerialNumber)

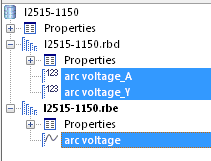
Suddlgshow("InductiveResistance",myFolders(3))

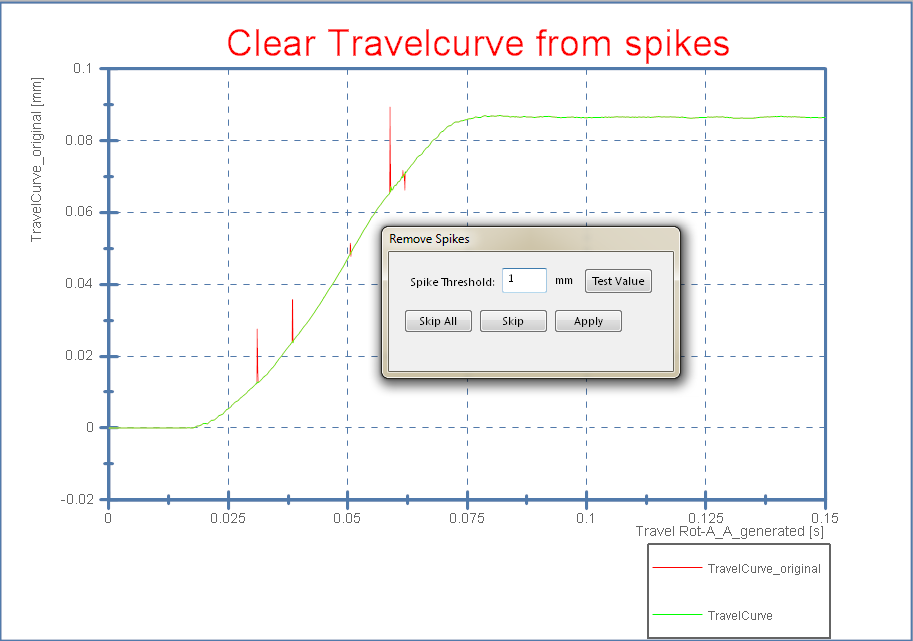


* Once the user is satisfied with the result and clicked apply, Export() is called and the export process starts.

Export():

This function goes through every shots specified and loads the raw data in the server and export all selected channels according to specification.

* For every shot it calls CreateEquidistantChannels(ShotName,PowChannels,PowNumbOfChn,samplingRate) to create an equidistant channel from the recorded test measurement which consists of an x-channel and a y-channel(usually stored as \_A and \_Y) and CropChannels(eqGroup,PowChannels,PowNumbOfChn,duration,samplingRate) to crop the channel according to the trip signal (named VS or TO).
* The travel curve is first transformed from degree to meter if the measurement data are in degrees. And then ShowSmoothDialog(eqGroup,PowChannels(j)) is called, the following dialog box is displayed.

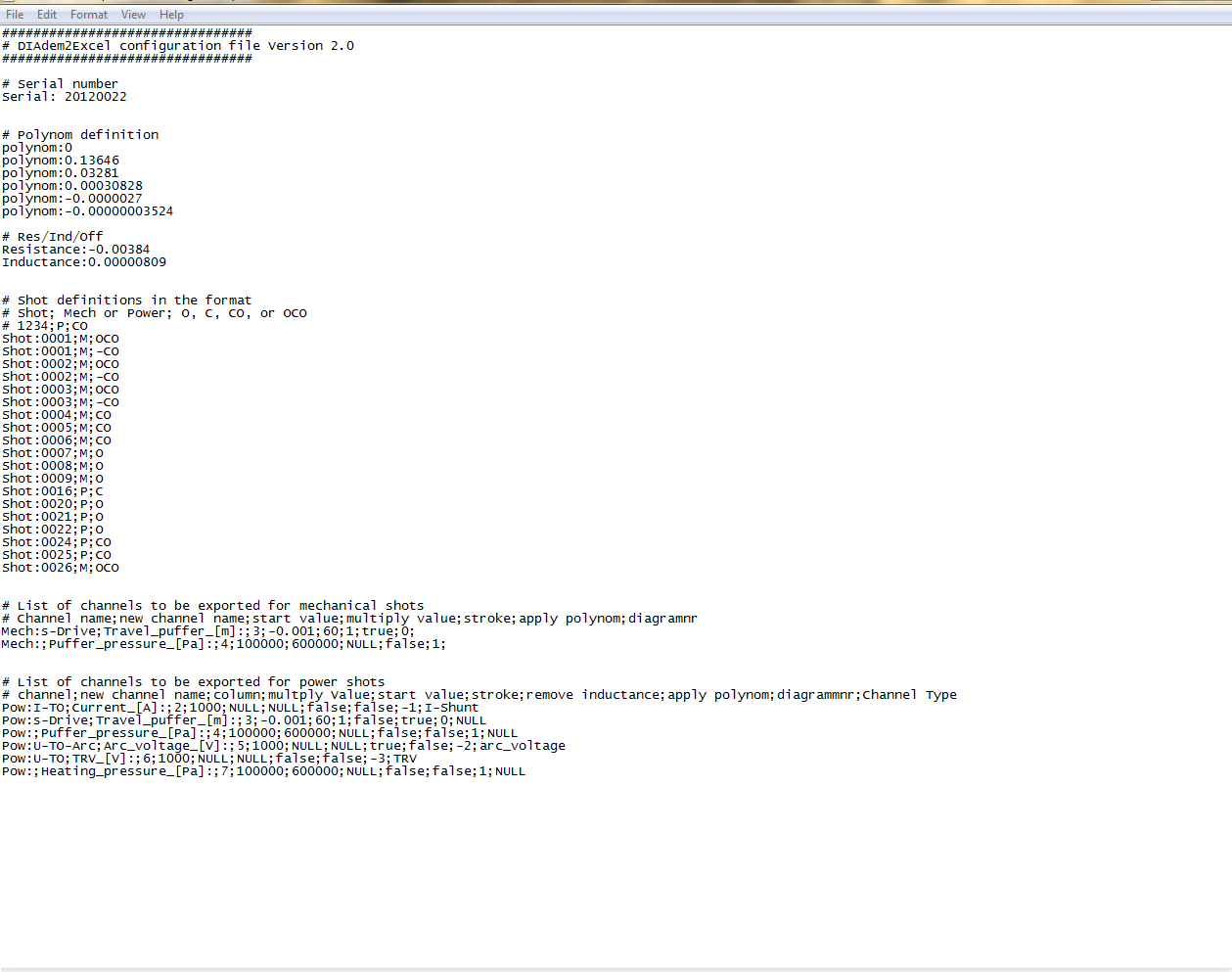


* For power shots, the arc voltage is filtered according to the inductance and resistance value determined initially by applying this function RemoveInductivResistantParts()

LoadCFG(cfgFile)

This function parses the input parameters from the configuration file

* The configuration file contains all the necessary parameters to extract and filter the data. These data are stored in arrays.

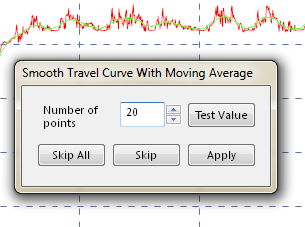
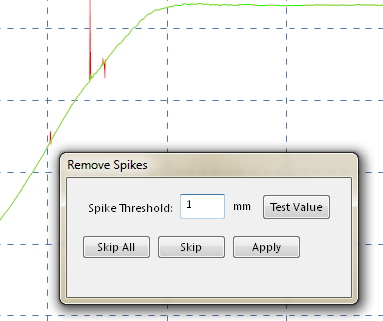


* At the end it also detects the name of the trigger channel (VS or TO)

ShowSmoothDialog(ChannelGroup, ChannelName)

This function is called from Export() to allow the user to smooth the travel curve. The dialog boxes called are in ExportTestWind.sub

* First ClearTravelSpikes(…) is called remove any point that is vastly different from the two adjacent points. This removes all large spikes in the travel curve.
* Then ChnSmooth(…) is called to take a running average of the channel. This clears any small oscillation that still remains.



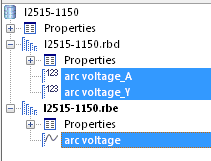
*ExportTest\_functions.vbs*

CreateEquidistantChannels(…):

The purpose of this function is to create equidistant channels from the measurement data which consist of an x-channel (indicated as \_A, \_AB or \_ABA at the end) and a y-channel (indicated as \_Y at the end)

The ending of the x-channel indicates the sampling rate for the data. For example \_AB means lower sampling rate at the beginning and higher at the end; and \_ABA means high sampling rate in the middle. The purpose of this is to get more refined data at the part of the test which is most interesting. Please ask Marco Mailand for a more in depth explanation if interested.

This function records all the names of the interested channels and pass them into equix(…) to create equidistant channels.



RemoveInductivResistantParts(…):

This function filters the influence of the intrinsic resistance and inductance of the circuit breaker by using the following two equations:

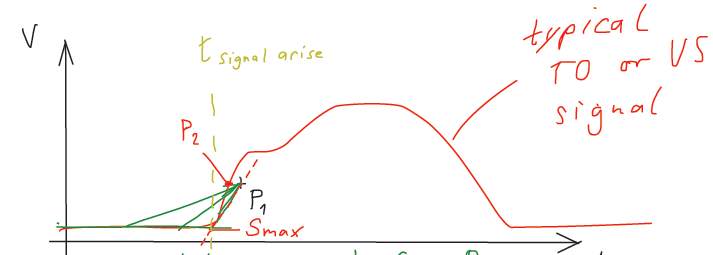
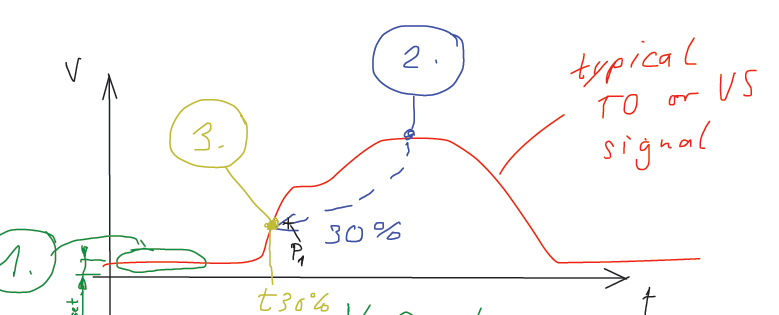
v_{\text{R}} \left( t \right) = {i_{\text{R}} \left( t \right)}R

v_{\text{L}}(t) = L \frac{\operatorname{d}i_{\text{L}}(t)}{\operatorname{d}t}

CropChannels(…):

This function detects the first rise from the trip signal and cut all the channels according to this time to focus on the interesting part.

An algorithm is used to find the first rise of the trip signal. the following picture is a graphical representation of the algorithm. For more information please ask Marco Mailand.



**Concept Tree**