## Installing Glue to MATLAB 2

## Connecting to the ETRE and VTRE databases 5

## Uninstalling Glue from MATLAB 11

## Settings File 13

## Part 1065 Assumptions 16

## Part 1065 Chemical Balance 17

## Part 1065 Determination of Dew Temperature from Water Concentration 22

## Determination of the Internal Tunnel Wall Temperature from the Measured External Wall Temperature and Ambient Air Temperature 24

## Channels Object Documentation 26

## Channel Name Documentation 27

## Channel Class Documentation 28

Analyzer Channel Class 30

Options Channel 32

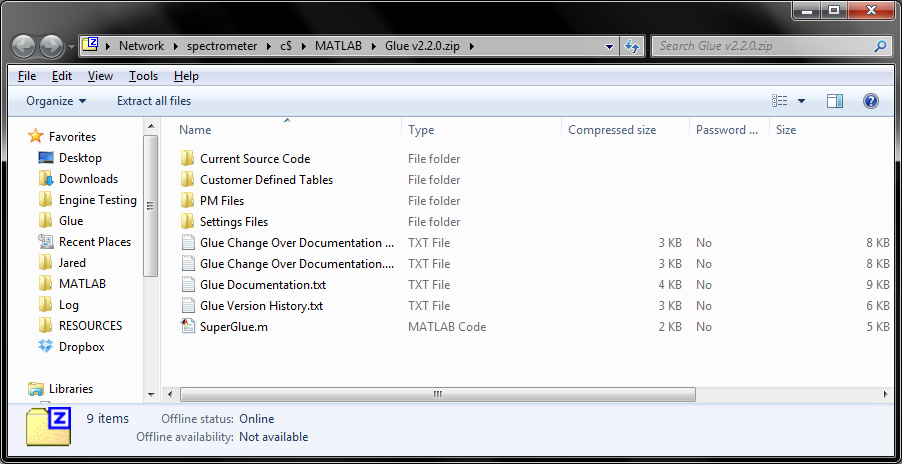
## Installing Glue to MATLAB

To install Glue to MATLAB follow these directions:

1. Place the ZIP file containing Glue where you’d like it to be installed. On SPECTROMETER and LAPTOP50 this location is C:\MATLAB\.

2. If there is no location C:\Log\ one must be created. This is log files are written to this location by Glue.

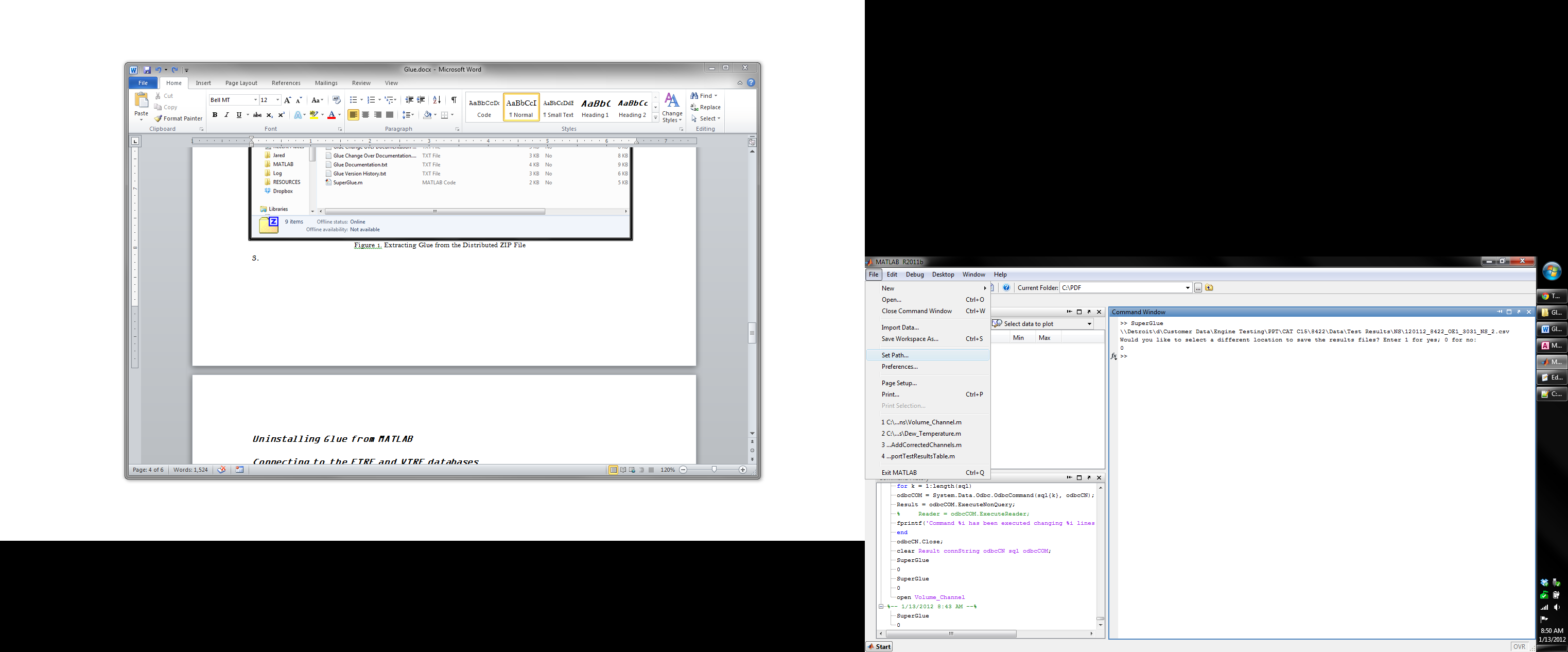
3. Extract all the files in the ZIP file. This is best done using a file compression program such as 7-Zip (which only requires a right click on the file); but it can be accomplished with Windows Explorer. To extract Glue using Windows Explorer double click the ZIP file to open it, on the second toolbar from the top click the button that says “Extract all files”. This will move all of the files into a new directory specifically for this installation of Glue.



Extracting Glue from the Distributed ZIP File

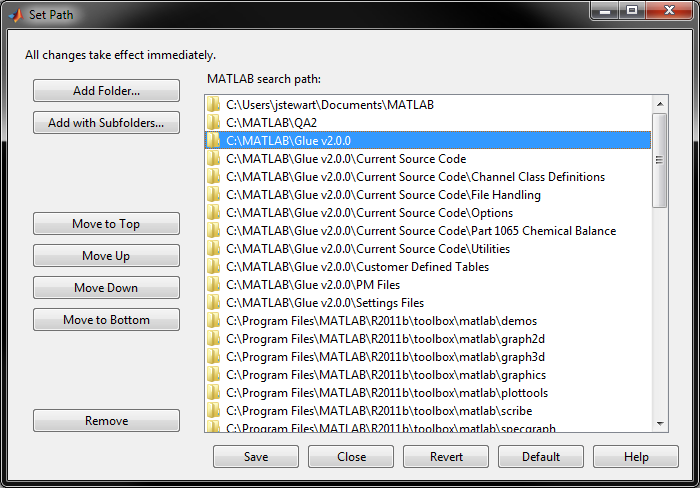
4. Launch MATLAB with Administrative Privileges. In MATLAB 2010 and before (SPECTROMETER), this is done by holding Control+Shift when starting the program, a prompt will ask if you would like to allow MATLAB to make changes to the computer. In MATLAB 2011 and later this is not necessary; MATLAB will prompt you for privilege escalation when you need to make changes.

5. In MATLAB select File >> Set Path. This will allow you to change the paths that MATLAB looks at when searching for .m files.

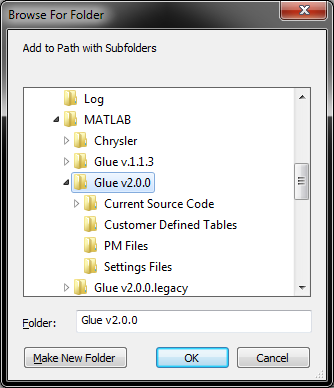


Changing the paths on which MATLAB looks for .m files.

6. If there is an existing installation of Glue it must first be uninstalled, follow the instructions later in this section to uninstall the previous version of Glue. Press the “Add with Subfolders” button and navigate to the location where you placed the extracted zip file – on both SPECTROMETER and Laptop50 this will be at C:\MATLAB\. Select the top directory and press “OK”



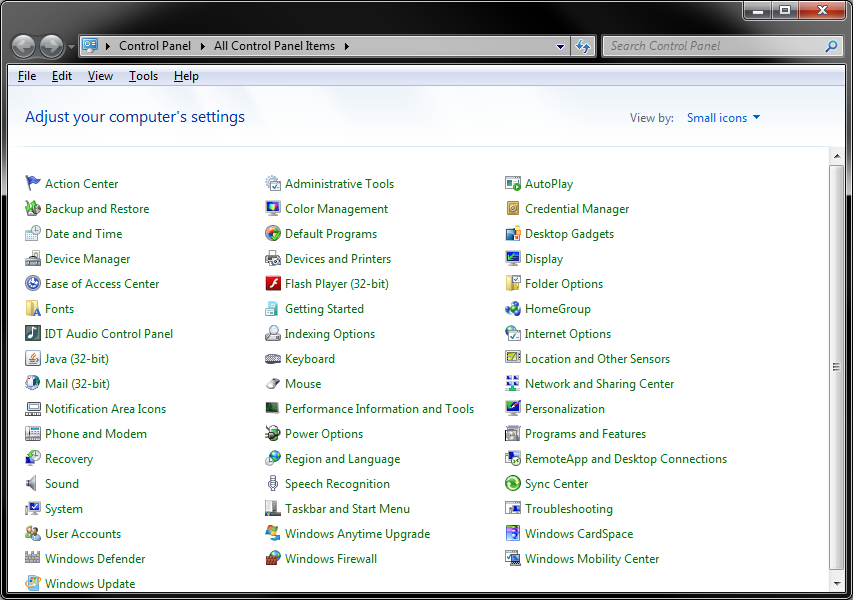
Choose the Add with Subfolders option from the Set Path menu.



Select the root directory of the Glue installation.

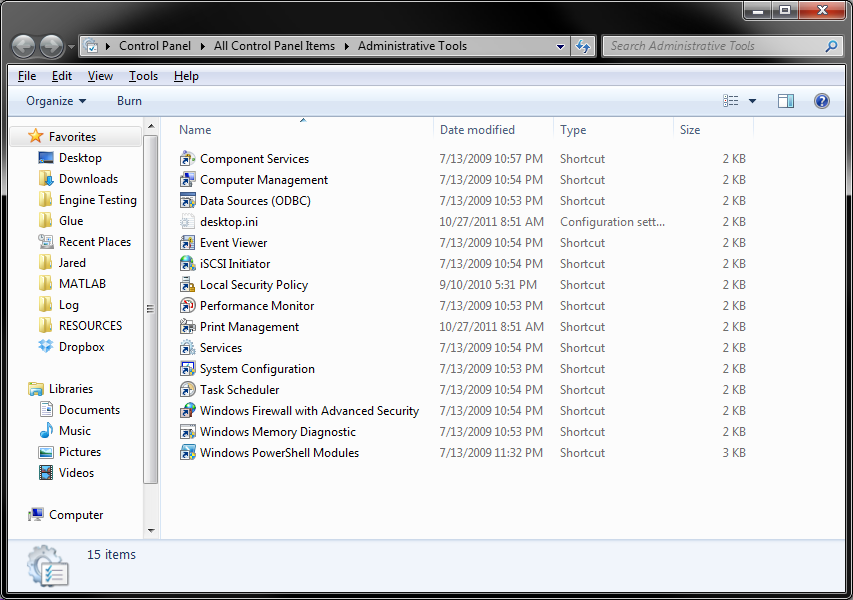
## Connecting to the ETRE and VTRE databases

1. Glue connects to the ETRE and VTRE databases to acquire more information concerning tests, however this is not necessary to make Glue operable. To connect to both these databases you need to use the Administrative Tools found in the Control Panel. If you do not see an Administrative Tools icon you can find it by changing from a Category view, to large or small Icons in the top right of the control panel.



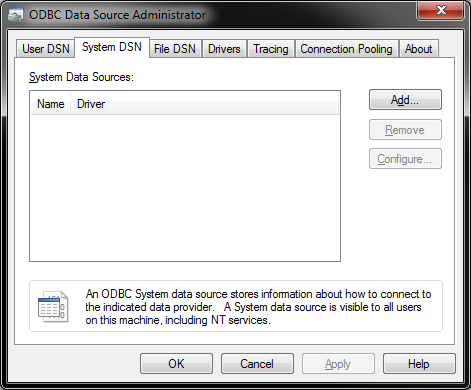
Select Administrative Tools from the Control Panel.

2. In the Administrative Tools select “Data Sources (ODBC)”. This will open the ODBC Data Source Administrator Panel.



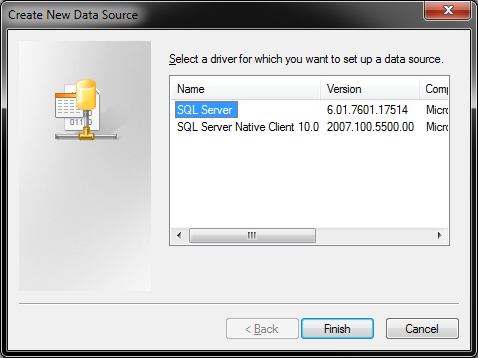
Select Data Sources (ODBC) from the Administrative Tools menu.

3. In the ODBC Data Source Administrator move to the System DSN (Data Source Name) Tab and then press the “Add…” button.



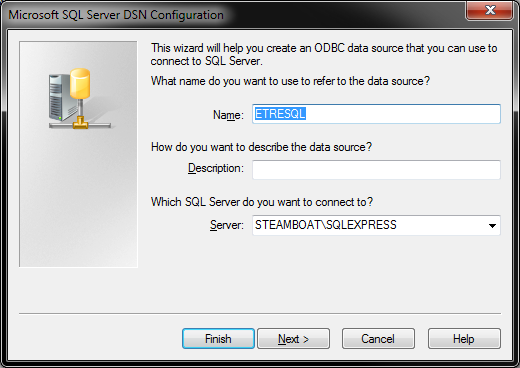
Navigate to the System DSN tab and choose to Add a new DSN.

4. The correct driver for the ETRE and VTRE databases is SQL Server, this will be in a list of available drivers.

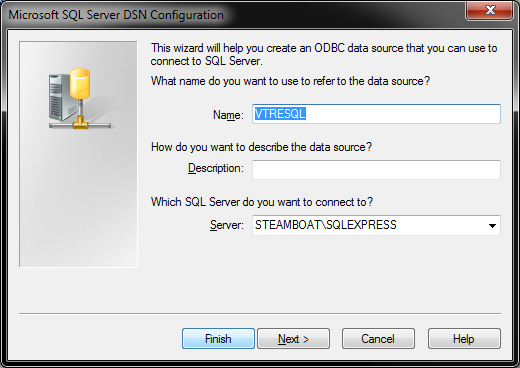


Select SQL Server from the list of available drivers.

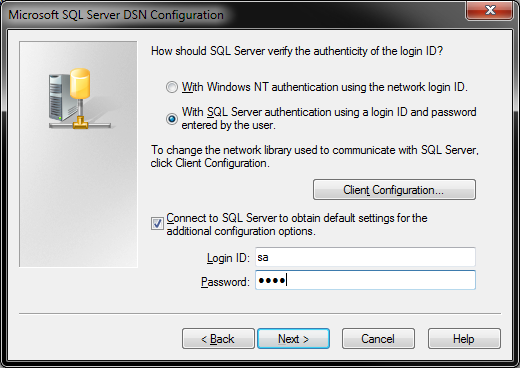
5. At this stage you will be able to name the database as you wish. In order for Glue to be able to access the databases they **must** be named ETRESQL for the engine database, and VTRESQL for the vehicle database. Glue is strictly an engine data post processing utility, however the PM results are being stored on the vehicle database. The server for both databases is STEAMBOAT\SQLEXPRESS.



Naming the engine database (ETRE).

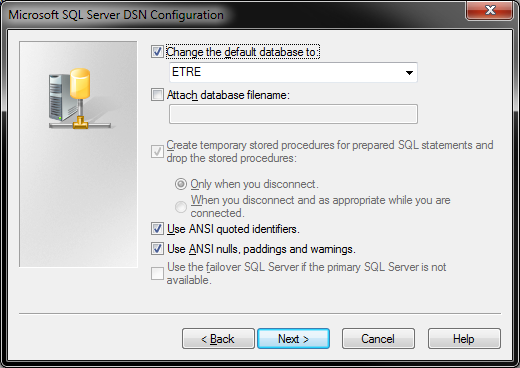


Naming the vehicle database (VTRE).

6. At this point it will be necessary to login to the STEAMBOAT\SQLEXPRESS server. To do this, change the authentication from Windows NT to SQL Server. The login ID is sa; the password is vtre. If you are not able to access the STEAMBOAT\SQLEXPRESS server this step will fail and you will not be able to complete the process.

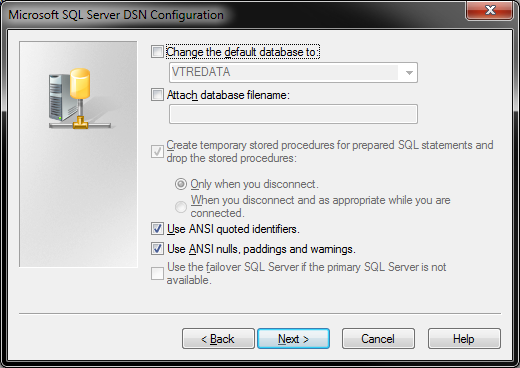
Logging in to STEAMBOAT\SQLEXPRESS with login ID sa; password vtre.

7. Change the default database to ETRE for the engine database or VTREDATA for the vehicle database.



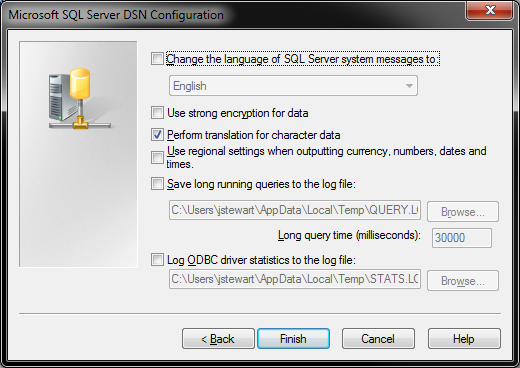
Select the default database as ETRE for the engine database.

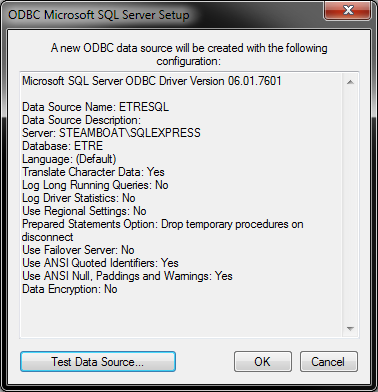
Select the default database as VTREDATA for the vehicle database.



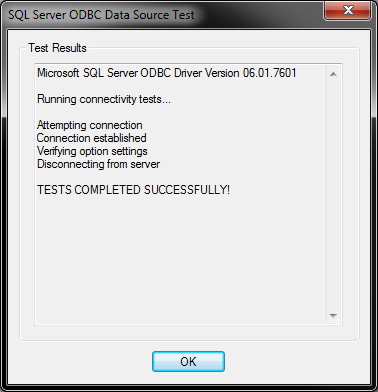
8. Press finish to complete the process, a screen will appear showing the DSN settings. Press the “Test Data Source…” button to verify that the connection is correctly set up.

Final screen of the Server DSN Configuration process.





Summary of the DSN configuration, press the Test Data Source button to verify connection.



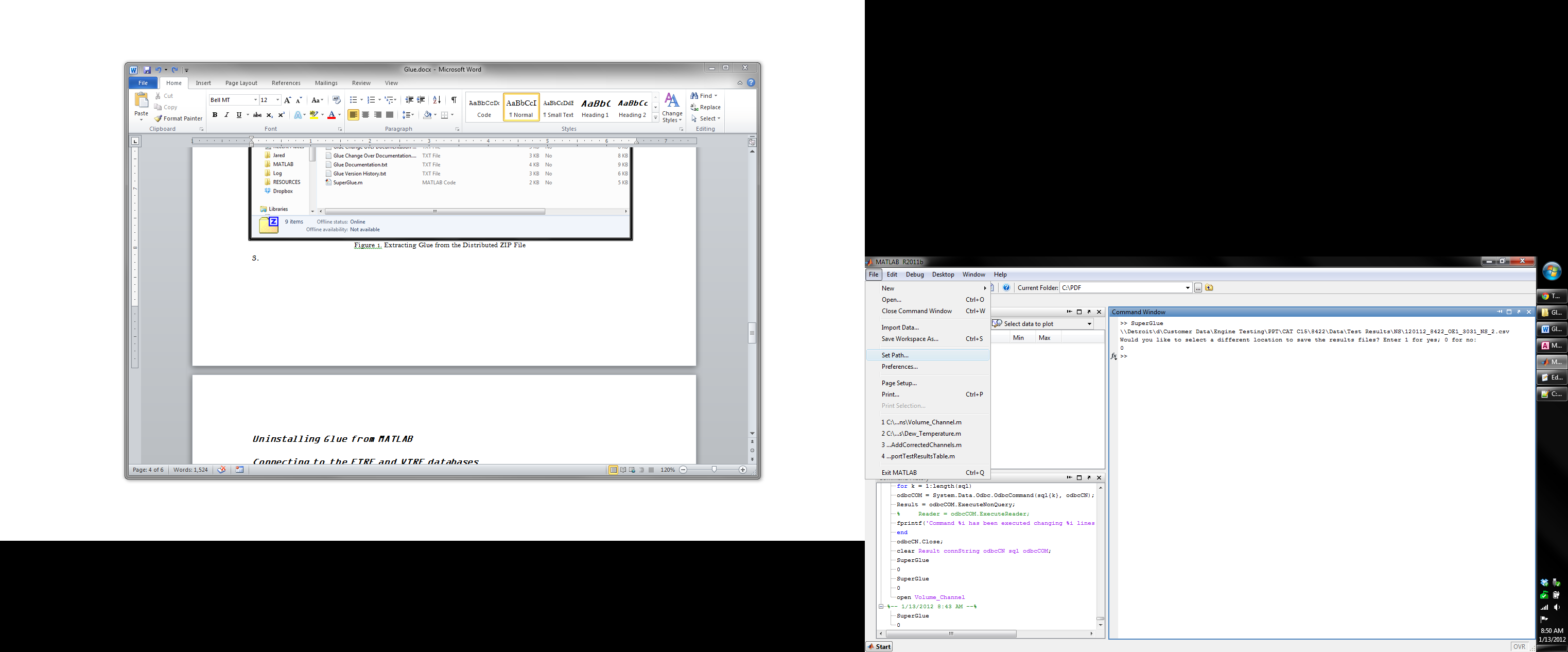
Connection successfully verified, DSN has been properly setup.

## Uninstalling Glue from MATLAB

When a new version of Glue is released it is important to uninstall the previous version of Glue from MATLAB so that an old .m file, without updated code, is not accidentally used. This itself is probably best done just before installing the new version of Glue in the same instance of MATLAB – so that it is not necessary to launch the program with Administrative Privileges twice.

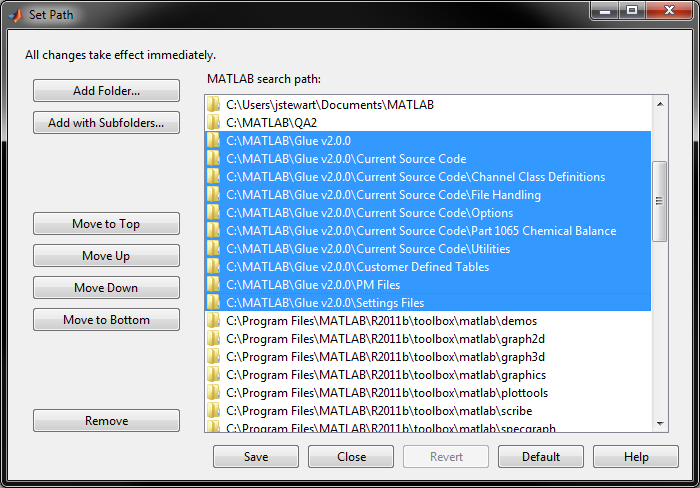
1. Launch MATLAB with Administrative Privileges. In MATLAB 2010 and before (SPECTROMETER), this is done by holding Control+Shift when starting the program, a prompt will ask if you would like to allow MATLAB to make changes to the computer. In MATLAB 2011 and later this is not necessary; MATLAB will prompt you for privilege escalation when you need to make changes.

2. In MATLAB select File >> Set Path. This will allow you to change the paths that MATLAB looks at when searching for .m files.



Changing the paths on which MATLAB looks for .m files.

3. Select all of the Glue folders and press the “Remove” button. This will completely uninstall Glue.



Remove all of the Glue folders.

## Settings File

The Settings File controls the behavior of Glue and also the parameters used in the calculations. Glue uses multiple Settings Files, the first is in the same set of folders which Glue is installed in; this is the local Settings File. The local Settings File either directs Glue to use the Engine Cell specific Settings Files on the Network, or to use the customized Settings in the local Settings File.

Some of the parameters listed are currently defunct; these parameters are either being updated or they are left in to be used as an example.

# Option\_Panel Network

# Network\_Path

# //Detroit/d/Customer Data/Engine Testing/Test\_Cell\_Management/Glue/

# Part\_1065.Ignition\_Type Compression

# Part\_1065.IsOn 1

# Part\_1065.CO2\_dil Ambient

# Part\_1065.CO2\_int Ambient

# Part\_1065.O2CO2\_intdry 0.209820

# Part\_1065.KH2Ogas 3.5

# RF\_CH4 1.0000

# RFPF\_C2H6 0.0

# Range\_Change\_Level 0.8

# Bag\_Dilute\_Ranges\_Locked 1

# ISO\_8178.IsOn 0

# ISO\_8178.Combustion Complete

# ISO\_8178.Fuel\_And\_Air\_Flow Off

# ISO\_8178.Air\_Flow Off

# ISO\_8178.Fuel\_Flow On

# ISO\_8178.GasPressure On

# ISO\_8178.MolarMass Off

# Number\_CDRs 0

# CDR\_1 SpindtAnalysis.txt

# CDR\_1\_Location Network

# Part\_1065.DriftCorrection 1

# Part\_1065.HCContaminationCorrection 1

# ModalWeighting.Type\_8M,1,1,1,1,1,1,1,1

# Change\_File\_Location SilenceYes

# Produce\_MAT\_File Ask

# Delete\_Modes SilenceNo

**Option\_Panel**: can be set to either Network or Local depending on user preference

**Network\_Path**: is the designated location of the files that need to be brought into Glue when Option\_Panel is set to Network, keep in mind that the local Option\_Panel governs the operation. This means that the local file will always control whether or not the program operates on the network, and where the program looks to.

**Part\_1065.Ignition\_Type**: Either spark or compression, this adjusts calculation of a few parameters in the Part 1065 chemical balance, namely the NOx correction and how NOx is subdivided into NO and NO2.

**Part\_1065.IsOn**: This allows the test, in particular coming from OE1 and OE2, to skip Part 1065 calculations in a stable and predictable manner.

**Part\_1065.CO2\_dildry**: This is the amount of CO2 in the dry dilution air, it is used in the chemical balance; currently this is the measured ambient value. The user may change this to a particular value, the value recommended in part 1065 is 0.000375

**Part\_1065.CO2\_intdry**: This is the amount of CO2 in the dry intake air, it is used in the chemical balance; currently this is the measured ambient value. The user may change this to a particular value, the value recommended in part 1065 is 0.000375

**Part\_1065.O2CO2\_intdry**: This is the amount of O2 and CO2 in the dry intake air, this is the value recommended by the Part 1065.

**Part\_1065.KH2Ogas**: is the water-gas reaction equilibrium coefficient. This is set to the Part1065 recommendation of 3.5.

**Part\_1065.DriftCorrection**: Set to 1 to turn on drift correction per Part 1065, a value of 0 turns this off and Part 1065 Chemical Balance calculations are done using Part 86 concentrations

**Part\_1065.HCContaminationCorrection:** Set to 1 to turn on HC Contamination Correction per Part 1065, a value of 0 turns this off a Part 1065 Chemical Balance calculations are performed without taking this into account.

**RF\_CH4**: This is the response factor of the hydrocarbon analyzer to methane; should be close to one - and this will vary per test cell.

**RFPF\_C2H6**: This is the response factor/penetration factor of the methane analyzer to ethane; should be close to zero - this will vary by test cell

**Range\_Change\_Level**: Is the percentage of maximum concentration where the analyzer is kicked into the next range. The current value is set at 80%.

**Bag\_Dilute\_Ranges\_Locked**: When a Cert Test isn't being run and this is set to No then the ranges aren't locked and can change at each mode. This influences the way that the drift correction is performed.

**ISO\_8178.IsOn**: This is currently deprecated - when the ISO calculations are completed this will be available for subsonic testing.

**ISO\_8178.Combustion**: This is currently deprecated

**ISO\_8178.Fuel\_And\_Air\_Flow**: This is currently deprecated

**ISO\_8178.Air\_Flow**: This is currently deprecated

**ISO\_8178.Fuel\_Flow**: This is currently deprecated

**ISO\_8178.GasPressure**: This is currently deprecated

**ISO\_8178.MolarMass**: This is currently deprecated

**Number\_CDRs**: The number of customer defined reports. While this isn’t a fully functional feature it does allow the user to define customer data presentation for the composite report.

**CDR\_N**: File name for the Nth CDR.

**CDR\_N\_Location**: Network or Local, where the CDR is stashed.

**ModalWeight.Type\_[TestType]**: Currently only available for SET, 8M, 13M, and ESC test types; this provides an automatic modal weighting calculation using the factors given. Be careful, the number of modes must match exactly, and the weights for any void modes – or modes that you don’t want included in the weighted calculation should be set to 0.

**Change\_File\_Location**. This can be used to silence user queries to change the save location of the Composite and Streaming reports. SilenceNo will shut off the query causing all files to be saved to the current location of the input file. SilenceYes will shut off the query and automatically provide a file dialog so the user can change the destination of the Composite and Streaming reports. Ask will prompt the user before deciding what to do.

**Produce\_MAT\_File**. This can be used to silence user queries to change the save location of the Composite and Streaming reports. SilenceNo will shut off the query and not produce a .MAT file for post processing. SilenceYes will shut off the query and automatically provide a file dialog so the user can change the destination of the Composite and Streaming reports. Ask will prompt the user before deciding what to do.

**Delete\_Modes**. This can be used to silence user queries to change the save location of the Composite and Streaming reports. SilenceNo will shut off the query causing no modes to be deleted during the QA setup. Ask will prompt the user to select modes for deletion before producing the QA report.

## Part 1065 Assumptions

1065.650(c)(1)(i) Correct for initial concentration on HC and CH4 by 1065.660(a); this includes dilution air background concentrations

HC is corrected, , as per 1065.660(a) Eq. 1065.660-1

CH4 is corrected, , as per 1065.660(a) Eq. 1065.660-1

The initial contamination factor is described in 1065.520(g)(5) as the concentration from overflow zero air flows. These values are limited to a maximum of 2% of the flow weighted mean wet (measured or expected) or 2 μmol/mol as per 1065.520(g)(7). This correction can be performed before or after chemical balance procedures in Glue using the Part1065.HCContaminationCorrection argument in the Settings File; a value of one tells Glue to perform the correction before the chemical balance, and a value of zero tells Glue to perform the correction after the chemical balance.

1065.650(c)(1)(ii) Perform the dry concentration to wet concentration conversion by 1065.659; this includes dilution background concentrations

All raw concentrations are corrected for water removed upstream as per 1065.659. We assume that the water removal process removes 100% of the water. This contrasts 1065.659(b) which suggests that the water downstream of these analyzers be measured. The appropriate conversion when the water concentration is measured is

The components NO and NO2 of measured NOx are assumed to have the following distributions in Part 1065.655(c)(1).

For Compression Ignition Engines

For Spark Ignition Engines

## Part 1065 Chemical Balance

The Part 1065 Chemical Balance calculations are organized into tiers. Each tier of calculations can be performed simultaneously; each subsequent tier depends on calculations performed in the previous tiers. This is part of a Newton-Raphson calculation to determine the quantities, bleep and bleep. The calculations performed in the final tier are used in place of initial guesses and subsequent calculations in the first tier.

Tier 1 Formulas

(1065.655-1)

(1065.655-2)

(1065.655-11)

(1065.655-13)

Tier 2 Formulas

(1065.655-6)

(1065.655-9)

(1065.655-10)

(1065.655-12)

(1065.655-14)

(1065.655-15)

(1065.655-16)

(1065.655-17)

(1065.655-18)

Tier 3 Formulas

(1065.655-4)

Tier 4 Formulas

The calculations for the carbon products from combustion, , and the amount of intake air required to produce actual combustion products, , feature a circular reference which must be removed

(1065.655-3)

(1065.655-7)

The solution for after the cyclic reference to is removed

()

Tier 5 Formulas

(1065.655-7)

Tier 6 Formulas

This tier evaluates the formulas for the iterated quantities, if the initial values are close to the true value then these values will be similar, otherwise they may be completely different

(1065.655-5)

(1065.655-8)

Tier 7 Formulas

The derivative of the dry emissions with respect to the water in the dry exhaust (there is no raw gas derivative) is

()

Tier 8 Formulas

The derivatives of the amount of hydrogen in dry exhaust with respect to both water and raw gas in the dry exhaust are

()

()

Tier 9 Formulas

The derivatives of the carbon products from combustion (using Equation 1 not 1065.655-3) with respect to both water and raw gas in the dry exhaust are

()

()

Tier 10 Formulas

The derivatives of the amount of intake air required to produce actual combustion products with respect to both water and raw gas in the dry exhaust are

()

()

Tier 11 Formulas

The net evaluation of and its associated derivatives with respect to both water and raw gas in the dry exhaust are

()

()

()

Tier 12 Formulas

The net evaluation of and its associated derivatives with respect to both water and raw gas in the dry exhaust are

()

()

()

Tier 13 Formulas

Solution of the system of equations is done using Cramer’s rule

()

()

()

()

Tier 1065.14 Formulas

Tier 1065.14 occurs just as the iterative chemical balance has finished. Because this chemical balance procedure assumes that the molar flow rate of the raw exhaust is know it is possible to calculate the following using Part 1065.

(1065.655-20)

(1065.655-22 in Working Document 7-18-2011)

(1065.655-21)

()

## Part 1065 Determination of Dew Temperature from Water Concentration

Part 1065.140 dictates that we need to monitor internal tunnel wall temperatures and the dew temperature of the diluted exhaust to ensure there is not a significant amount of water lost in the diluted exhaust. The internal tunnel wall temperatures were not found to be significantly different than the external tunnel wall temperatures (which we measure instead) at our facility through a [heat transfer study](#_Determination_of_the). As we do not measure the dew temperature it is necessary to calculate it using the chemical balance procedures provided by Part 1065.655. The chemical balance yields the concentration of water in the diluted exhaust; Dalton’s Law extends this to the water vapor pressure.

Part 1065.645 uses the Goff-Gratch equations to estimate the saturation water vapor pressure at a given temperature – this can be used with the pressure measured and the dew temperature to find the vapor pressure for unsaturated conditions. However, this equation is not simple to find the inverse of (determining the dew temperature for a particular water vapor pressure). As such a nonlinear regression has been used to approximate the inverse equation. The regression is valid for a range of temperatures -50 °C to 100 °C.

The regression takes two steps, the first step approximates the curve and the second step reduces the average error from 0.3 °C to 0.1 °C. The coefficient of determination R2 for the combined model is 0.999993.

The first stage of calculation

()

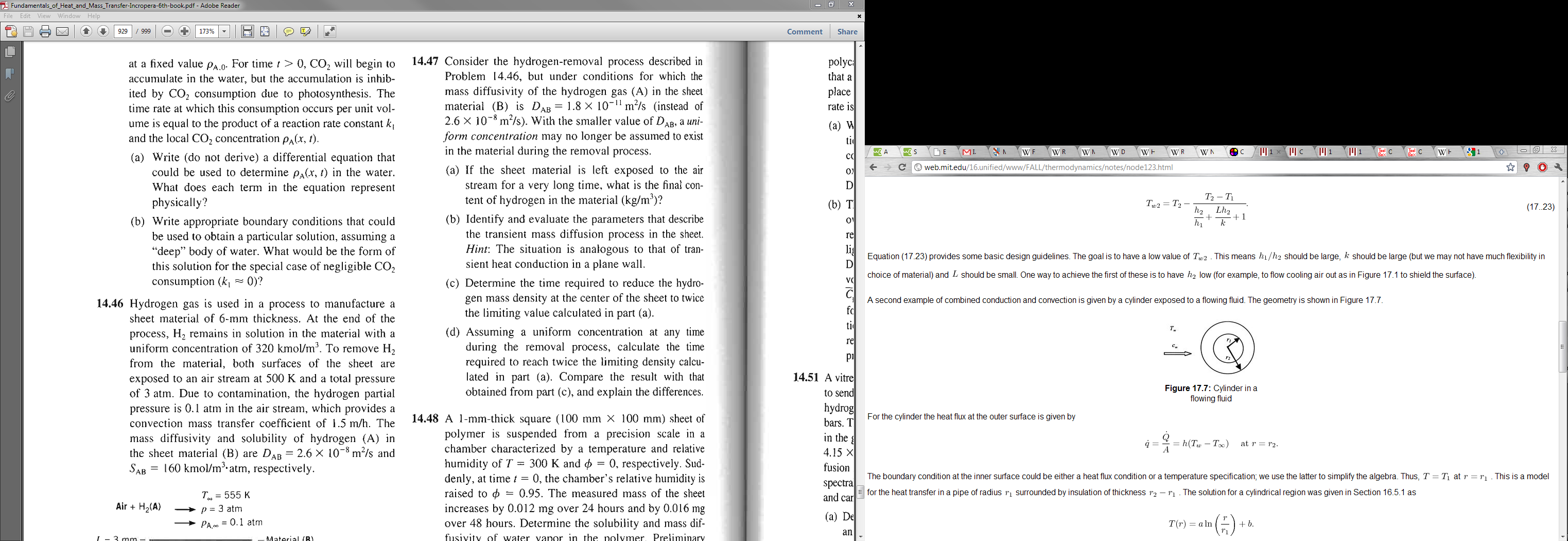
The second stage of calculation enhances the value calculated in the first stage

()

## Determination of the Internal Tunnel Wall Temperature from the Measured External Wall Temperature and Ambient Air Temperature

This solution was modified from the results produced here (<http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node123.html>)

which considers a cylinder in a flowing fluid.



The described problem relates the temperature of the cylinder at a specific radius, *r*, for a known inner wall temperature as

Solving for the boundary condition yields

Additionally, from

and

Therefore

and

Substituting in so that equation is solving for , and then rearranging so that the temperature gain, , is expressed as a function of the difference between the measured temperature and the ambient temperature, , yields

Given ; ; ; and (approximately the conditions of our CVS). The difference between internal and external wall temperatures is 0.014 times the difference between the external wall temperature and the ambient temperature.

## Channels Object Documentation

The datastream variable in SuperGlue is a Channels object. The Channels object inherits from the containers.Map object which in turn inherits from the handle container. This has several implications;

(1) a particular channel object can be pulled out of the datastream variable with the following syntax

# myChannelObject = datastream(myChannelName);

where myChannelName is a string or a variable containing a string

(2) a retrieved channel object from the datastream variable is a singleton – that is any changes to the retrieved object are reflected in the original object

# myChannelObject = NULL;

# . . . datastream(myChannelName) . . .; %ERROR!

(3) Channel specific operations and variables can be addended to the retrieved object using the following syntax

# datastream(myChannelName).ChannelSpecificOperation();

# datastream(myChannelName).ChannelSpecificVariable;

this is a particularly useful feature if you want to access the same data, or perform the same operation, for several channels

furthermore, this is polymorphic, that is MATLAB can use the same function call to do different things – this is particularly useful if you want to bulk convert data, like changing all pressure measurements to mm.Hg

## Channel Name Documentation

The channel names are parsed in a very specific sense to remove any special characters, any units, and to correct for any spacing issues.

Characters replaced with subscores

# ! @ # $ % ^ & \* ( ) / \ [ ] ’ ” . , ° -

All spaces are replaced with subscores

The following analyzer name substitutions are made

Original Name Processed Name

# (l) \_l\_

# (Tr) \_tr\_

# (egr) \_egr\_

Anything else contained in parentheses is assumed to be a unit and removed from the channel name – the unit is preserved in the channel object

The string ‘Bench’ is removed from the channel name for succinctness.

Any leading or trailing subscores are removed, and any repeated subscores are removed.

MATLAB won’t accept leading numbers – in the event of a leading number a A\_ is added to the front of the channel name.

Examples:

Original Name Processed Name – actual accessor

# ‘CO(l)\_Bag/Dilute\_Bench (ppm)’ ‘CO\_l\_Bag\_Dilute’

# ‘CO(l)\_Bag/Dilute\_Bench (Grams)’ ‘CO\_l\_Bag\_Dilute’

# ‘DT\_Air\_2 (C°)’ ‘DT\_Air\_2’

# ‘12\_TC (C°)’ ‘A\_12\_TC’

# ‘INCA4000 (none)’ ‘INCA4000’

## Channel Class Documentation

The Channel super class is inherited by all specific Channel classes. The inheritance is what allows the Channels class (datastream) to contain a variety of different Channel types. The Channel class defines variables common to all channels and the class constructor.

Variables defined in Channel.m

Parent

Name

Type

IsOn

Current\_Units

StreamingData

ModeCompositeData

PhaseCompositeData

Index

Exported = 0;

Parent – just points to the Channels class (datastream) which holds the Channel object

Name – holds the original name (not the processed name) for the channel with units removed

Type – holds the type of Channel that this is

IsOn – boolean for on – deprecated **do not use**

Current\_Units – string holding the current units. Some units are corrected as they enter datastream if a known variation is observed, other units will not be corrected to a known unit.

StreamingData – holds second by second data for 1 Hz exports and 10Hz data for 10Hz exports, essentially original data – typically a double array

ModeCompositeData – holds composited data for each mode – typically a double array

PhaseCompositeData – unused

Index – initial location of data in csv file, **do not use**

Exported – labels a channel as having been in the standard export, prevents channels from appearing twice in the final reports, **do not use**

Most channel class definitions are roughly the same – deviations occur in any unit conversions that are provided in each channel class and in compositing functions.

There are several ways to apply a unit conversion to the channel

To convert the channel in an absolute manner so that the channel has the same units in future uses do the following:

# datastream(myChannel).ConvertChannel(NewUnit);

To convert the data only for this calculation, thus leaving the original channel intact do the following:

# . . . datastream(myChannel).ConvertStreaming(NewUnit) . . . ;

# . . . datastream(myChannel).ConvertModeComposite(NewUnit) . . .;

where NewUnit is a string or variable containing a string

Conversions available

ChannelName Acceptable Units

# Work kW.hr, hp.hr

# P\_ (Pressure Channel) mm.Hg, in.Hg, kPa, MPa, bar, mbar, atm, psi, in.H2O

# T\_ (Temperature Channel) °F, K, R, °C

# Q\_ (Flow Channel) scf/s, scf/m, scf/h, scm/s, scm/m, scm/h, mol/m, mol/s

# (Analyzer Channel) mol/mol, %, ppm,

# oz, lb, kg, g,

# \*kW.hr, \*/hp.hr

Typically the following commands will return a double array of data

# datastream(myChannel).StreamingData;

# datastream(myChannel).ModeCompositeData;

Exceptions to this rule exist on the following page

**The Analyzer Channel Class**

Also contains several additional variables exclusive to analyzers

# VZS

# Ambient

# Ranges

VZS is a containers.Map class where VZS(Index) will return the virtual zero span data as a structure for the range indicated using the Index variable – Index must be an integer, particular care must be exercised with this structure because you are not guaranteed any ranges! Sometimes you will have only range 3, other times you will have ranges 1, 2, and 3, and yet other times only range 1 will be available. VZS has the following structure

datastream(myAnalyzerChannel).VZS(Range).MaximumConcentration #FROM DATABASE

datastream(myAnalyzerChannel).VZS(Range).ReferenceZero #FROM DATABASE

datastream(myAnalyzerChannel).VZS(Range).ReferenceConcentration #FROM DATABASE

datastream(myAnalyzerChannel).VZS(Range).RawPreTestZero #FROM DATABASE

datastream(myAnalyzerChannel).VZS(Range).RawPreTestSpan #FROM DATABASE

datastream(myAnalyzerChannel).VZS(Range).RawPostTestZero #CALCULATED - USED IN APPLYING 1065 DRIFT CORRECTIONS

datastream(myAnalyzerChannel).VZS(Range).RawPostTestSpan #CALCULATED - USED IN APPLYING 1065 DRIFT CORRECTIONS

datastream(myAnalyzerChannel).VZS(Range).CorrectedPreTestZeroCheck #FROM DATABASE - USED IN APPLYING HC/NMHC/CH4 CORRECTIONS FROM PART 1065.660

datastream(myAnalyzerChannel).VZS(Range).CorrectedPostTestZero #FROM DATABASE

datastream(myAnalyzerChannel).VZS(Range).CorrectedPostTestSpan #FROM DATABASE

Ambient is a structure with the following variables

datastream(myAnalyzerChannel).Ambient.PreTest #CALCULATED

datastream(myAnalyzerChannel).Ambient.PostTest #CALCULATED

datastream(myAnalyzerChannel).Ambient.Average #CALCULATED

datastream(myAnalyzerChannel).Ambient.Part86PreTest #FROM DATABASE

datastream(myAnalyzerChannel).Ambient.Part86PostTest #FROM DATABASE

datastream(myAnalyzerChannel).Ambient.Part86Average #CALCULATED

datastream(myAnalyzerChannel).Ambient.Part1065PreTest #CALCULATED

datastream(myAnalyzerChannel).Ambient.Part1065PostTest #CALCULATED

datastream(myAnalyzerChannel).Ambient.Part1065Average #CALCULATED

Ranges is a double array of the same size as the mode composite data which identifies which analyzer range was used on which mode of the test – range locking can be applied in the settings file.

Other changes

The Current\_Units field has changed to a cell array to capture concentration units {1}, mass units {2}, and brake specific mass units {3}.

The Streaming Data and ModeCompositeData double arrays have been replaced with structures which return double arrays – **N.B. Mass and Concentration use uncorrected analyzer readings and is incorrect unless you are performing the Part 1065 drift validation, you should always use Part86Mass/Concentration or Part1065Mass/Concentration**

StreamingData ->EXPANDED STRUCTURE

-----------------------------------------------------------------

[NoVZS\_Correction]

datastream(myAnalyzerChannel).StreamingData.Concentration

[ECCS\_PreTestVZS\_Correction]

datastream(myAnalyzerChannel).StreamingData.Part86Concentration

[PreAndPostTestVZS\_Correction]

datastream(myAnalyzerChannel).StreamingData.Part1065Concentration [Part1065Calculation\_NoVZS\_Correction]

datastream(myAnalyzerChannel).StreamingData.Mass

[ECCS\_PreTestVZS\_Correction]

datastream(myAnalyzerChannel).StreamingData.Part86Mass

[Part1065Calculation\_PreAndPostTestVZS\_Correction]

datastream(myAnalyzerChannel).StreamingData.Part1065Mass

ModeCompositeData ->EXPANDED STRUCTURE

-----------------------------------------------------------------

[NoVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Concentration

[ECCS\_PreTestVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Part86Concentration

[PreAndPostTestVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Part1065Concentration [Part1065Calculation\_NoVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Mass

[ECCS\_PreTestVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Part86Mass

[Part1065Calculation\_PreAndPostTestVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Part1065Mass

[Part1065Calculation\_NoVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.BrakeSpecificMass

[Where appropiate - not currently supported]

datastream(myAnalyzerChannel).ModeCompositeData.MassPerMile

[ECCS\_PreTestVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Part86BrakeSpecificMass

[Where appropriate - not currently supported]

datastream(myAnalyzerChannel).ModeCompositeData.Part86MassPerMile

[Part1065Calculation\_PreAndPostTestVZS\_Correction]

datastream(myAnalyzerChannel).ModeCompositeData.Part1065BrakeSpecificMass

[Where appropriate - not currently supported]

datastream(myAnalyzerChannel).ModeCompositeData.Part1065MassPerMile

Conversion functions require a little bit more finesse as well

# datastream(myAnalyzerChannel).ConvertStreaming(NewUnit, DataSetToConvert);

# datastream(myAnalyzerChannel).ConvertModeComposite(NewUnit, DataSetToConvert);

NewUnit as explained above the \* in brake specific unit can be replaced with any of the mass quantities.

DataSetToConvert must be set to any of the following ‘Concentration’, ‘Mass’, ‘BrakeSpecificMass’ – BrakeSpecificMass is not available for StreamingData

**The Options Channel**

The options channel represents data which isn’t specific to any measurement and doesn’t change throughout the test. It is a structure without any of the standard conventions. The correct Channel name is ‘Options’. The typical data structure is as follows (keep in mind this is from a 2 mode test) variable names on left – representative values on right

# Option\_Panel: 'Network'

# Network\_Path: '//Detroit/d/Customer Data/Engine Testing/Test\_Cell\_Management/Glue/'

# Part\_1065: [1x1 struct]

# RF\_CH4: 1.1185

# RFPF\_C2H6: 0

# Range\_Change\_Level: 0.8000

# Bag\_Dilute\_Ranges\_Locked: 0

# ISO\_8178: [1x1 struct]

# Number\_CDRs: 0

# CDR\_1: 'SpindtAnalysis.txt'

# CDR\_1\_Location: 'Network'

# Test\_Start: '22-Jan-2013 07:24:12'

# Test\_End: '22-Jan-2013 08:28:15'

# Test\_Duration: '1 hr, 4 min'

# Test\_Cell: 'EW4'

# Start\_Eng\_Hrs: 3482

# Test\_Tech: 'SSPENCER'

# Customer: 'ISUZU'

# Engine\_ID: 'K1Q001'

# SP\_P\_SimBaro: 758

# SP\_T\_In\_Air: 25

# Particulate\_On: 'No'

# DPF\_On: 'No'

# DPF\_Regen: 'No'

# Start\_Type: 'Hot'

# Engine\_Calibration: '85-7-4-3\_SA19\_6H\_HCM33t\_SY2CSV\_201211206\_Q001\_EA\_pruge\_HIGHALT\_9Hot'

# PreTest\_Comment: '(n/a)'

# PostTest\_Comment: 'Mode 2 is CP. Void mode 1.'

# Test\_Type: 'CP'

# Sample\_Configuration: 7

# Aftertreatment\_Configuration: 0

# Fuel: [1x1 struct]

# REPS\_Setup\_File: '85-7-4-3\_SA19\_6H\_HCM33t\_SY2CSV\_201211206\_Q001\_EA\_pruge\_HIGHALT\_9'

# Cert\_Test\_Flag: 4864

# Engine\_Bench\_Sample\_Position: 0

# Mid\_Bench\_Sample\_Position: 1

# Flex\_Bench\_Sample\_Position: 5

# Tailpipe\_Bench\_Sample\_Position: 0

# Smoke\_Meter\_Sample\_Position: 0

# MSS\_Sample\_Position: 0

# FTIR\_Sample\_Position: 1

# Void\_Flag: 0

# Bag\_Dilute\_Bench: 1

# Engine\_Bench: 1

# Tailpipe\_Bench: 1

# Tailpipe\_CO\_l\_InUse: [1 1]

# Engine\_CO\_l\_InUse: [0 0]

# Bag\_Dilute\_CO\_l\_InUse: [1 1]

# Molar\_Mass: [1x1 struct]

# delta\_T: 1

# Test\_Name: '130122\_K1Q001\_EW4\_9259\_CP\_1'

# Use\_HC: 0

Some of the more interesting substructures are as follows

# datastream(‘Options’).Fuel

# Specific\_Gravity: 0.8453

# Lower\_Heating\_Value: 18449

# Name: 'CERT DIESEL 121228'

# w\_C: 0.8680

# w\_H: 0.1320

# w\_O: 0

# w\_S: 9.2690e-006

# w\_N: 0

# alpha: 1.8123

# beta: 0

# gamma: 4.0007e-006

# delta: 0

# datastream(‘Options’).Molar\_Mass

# HC: 13.8412

# CO: 28.0101

# CO2: 44.0095

# NOx: 46.0055

# CH4: 16.0400

# O2: 31.9989