

# Modern Programmable Petrophysics Software Workflows

By: *P.Geoph. Oscar G. Gonzalez*

**Geoloil LLC**, Software Director

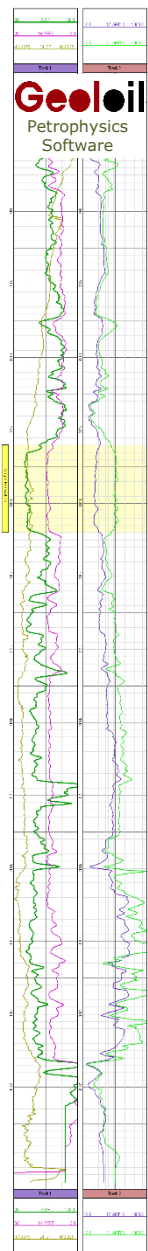
*Presented at SPWLA Houston 2025 Dec 5<sup>th</sup>. Technology Exhibition*

We introduce a programmable framework for applying user-defined petrophysical workflows to wells that share a common set of raw log curves. Each workflow is structured as a sequential chain of functions and scripts, organized row by row. The output of each row serves as the input for the subsequent row, enabling streamlined and modular data processing.

Tasks such as property computations, user-defined equations, algorithms, irregular depth curve stretching, curve merging, and calibration, are efficiently handled using a dedicated, domain-specific petrophysical scripting language. Once curve aliases and named parameters are established, the workflow can be seamlessly applied across a collection of wells, enabling immediate and consistent processing.

[geoloil.com](http://geoloil.com)



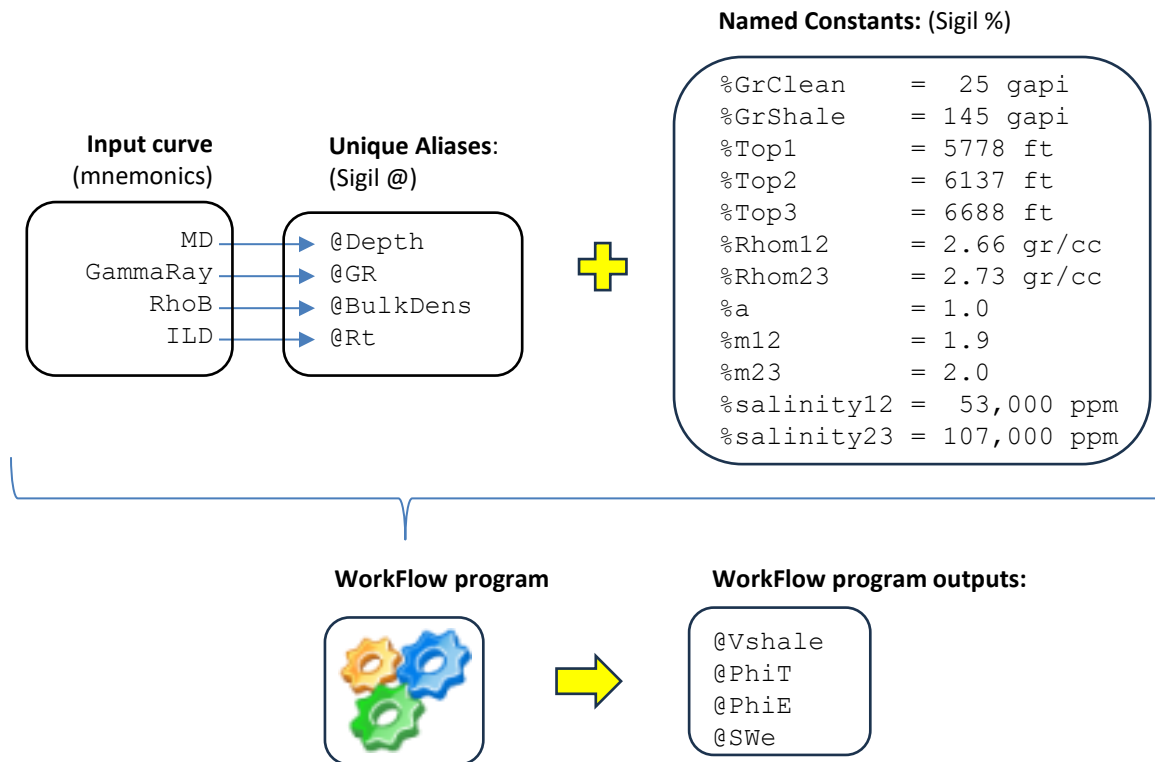


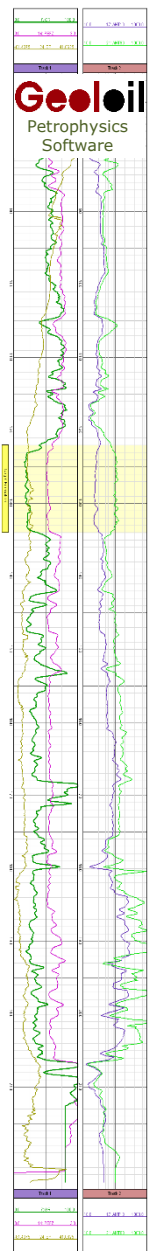
# What is a workflow program?

*“A **workflow** program is a chained sequence of functions and script programs applied to a particular set of raw input log curves, that yields a desired collection of output log curves.”*

Once a workflow is properly defined, it can be saved and applied to different wells with the same set of input curves, to produce final interpreted log curves. Each workflow entry data consists two elements:

1. **Curves aliases:** The original raw curve mnemonics are assigned to unique defaults, or user defined aliases.
2. **Named constants, or parameters.** As any constant parameter is easily converted to a curve with constant values per depth, the workflow program can handle variable parameters per zone as step-function constants using composed indicator functions {0,1}





# Anatomy of the workflow program



Sequential execution order ↓

```
Script : Indicator {0,1} Flag for Zone 1-2
Script : Indicator {0,1} Flag for Zone 2-3
Script : Staircase Salinity curve for whole Zone 1-3
Function : Borehole Temperature
Function : Vshale
Function : Rw variable curve from staircase salinity
Script : Staircase RhoM curve for zone 1-3
Function : Total Porosity curve from variable RhoM curve
Function : Effective Porosity curve
Script : Staircase mCurve for zone 1-3
Function : SWe Effective water saturation curve
```

WorkFlow program outputs:

```
@Vshale
-
-
@PhiT
@PhiE
-
@SWe
```

This “meta-program” is a chained, sequential collection of 11 functions and scripts: A collection of programs behind the scenes.

Since all the input information uses aliases and named constants, it can be saved and applied to different wells that use the same collection of original raw input curves.

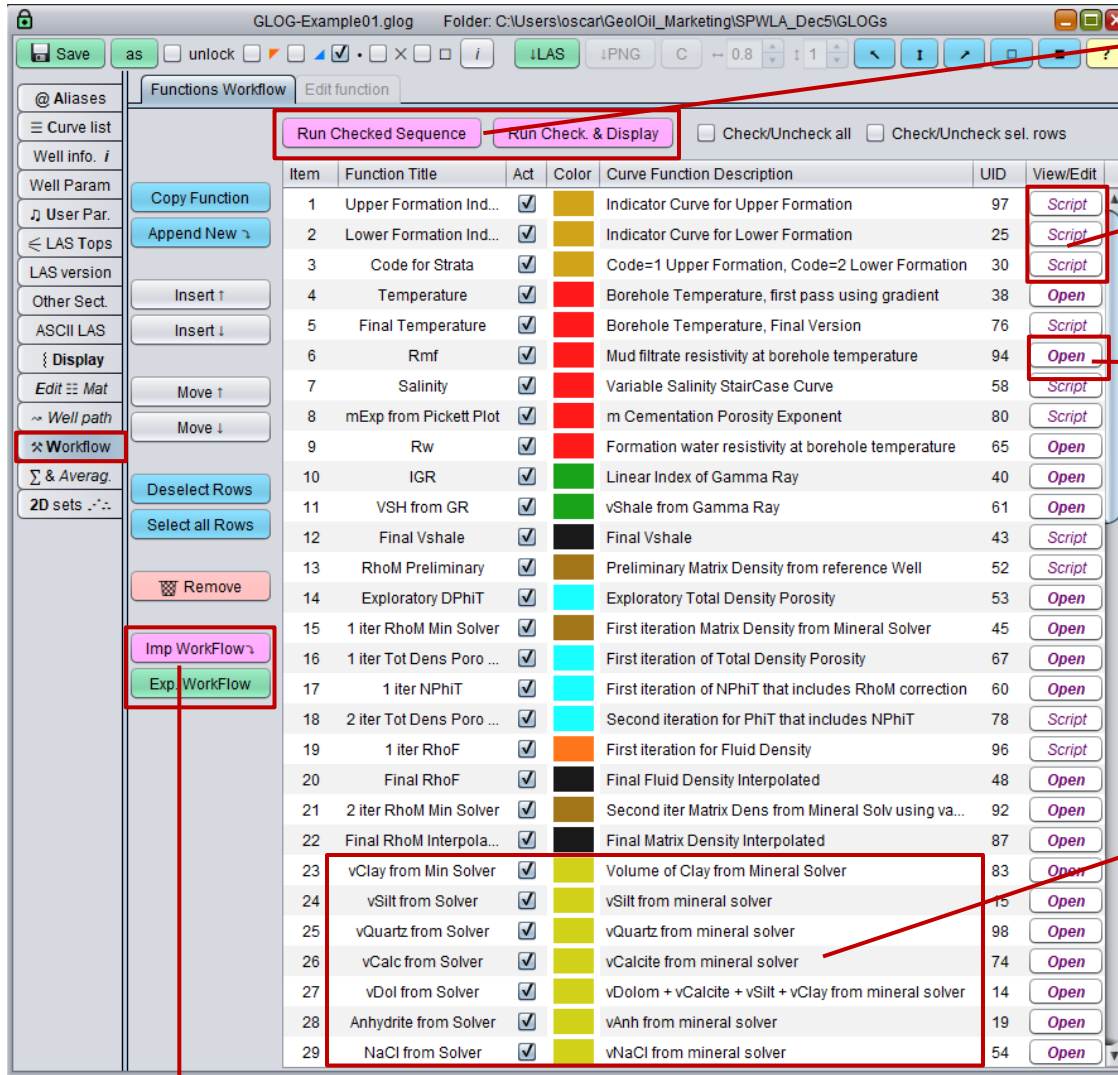
## Modern Math & Computer Science Operations

The internal computations engine that drives functions and scripts uses state of the art math:

- Ordinary scalar algebra
- Matrix algebra
- 4 states tetra-lean logic algebra: Non-applicable, Unknown, True, False. More complex than the standard Boolean two states algebra.
- Fuzzy logic algebra. A continuum measure of “belonging” to a result:  $0 \leq f \leq 1$  (Partially false to partially true). Not to be confused with probability.
- Behind the scenes, each script is internally a program that automatically writes a program: A Compiler.
- Occasional Parallel Programing, when needed to boost speed.
- This framework was carefully designed from the beginning without patches: AI and future ready tools.



## A Workflow program example



The screenshot shows the 'Functions Workflow' window in Geoloil. The window title is 'GLOG-Example01.glog' and the folder is 'C:\Users\losca\Geoloil\_Marketing\SPWLA\_Dec5\GLOGs'. The interface includes a toolbar with buttons like 'Save', 'as', 'unlock', 'LAS', 'PNG', 'C', and a 'Functions Workflow' tab. Below the toolbar, there are buttons for 'Run Checked Sequence', 'Run Check & Display', 'Check/Uncheck all', and 'Check/Uncheck sel. rows'. A table lists 29 items, each with a function title, an 'Act' checkbox, a color, a description, a UID, and a 'View/Edit' button. The 'View/Edit' buttons are either 'Script' or 'Open'. A red box highlights the 'Run Checked Sequence' and 'Run Check & Display' buttons. Another red box highlights the 'Script' buttons for the first three rows. A third red box highlights the 'Open' button for row 4. A fourth red box highlights the 'Imp Workflow' and 'Exp Workflow' buttons. A fifth red box highlights the rows 23 through 29, which are related to the mineral solver.

Item	Function Title	Act	Color	Curve Function Description	UID	View/Edit
1	Upper Formation Ind...	<input checked="" type="checkbox"/>	Yellow	Indicator Curve for Upper Formation	97	Script
2	Lower Formation Ind...	<input checked="" type="checkbox"/>	Yellow	Indicator Curve for Lower Formation	25	Script
3	Code for Strata	<input checked="" type="checkbox"/>	Yellow	Code=1 Upper Formation, Code=2 Lower Formation	30	Script
4	Temperature	<input checked="" type="checkbox"/>	Red	Borehole Temperature, first pass using gradient	38	Open
5	Final Temperature	<input checked="" type="checkbox"/>	Red	Borehole Temperature, Final Version	76	Script
6	Rmf	<input checked="" type="checkbox"/>	Red	Mud filtrate resistivity at borehole temperature	94	Open
7	Salinity	<input checked="" type="checkbox"/>	Red	Variable Salinity StairCase Curve	58	Script
8	mExp from Pickett Plot	<input checked="" type="checkbox"/>	Red	m Cementation Porosity Exponent	80	Script
9	Rw	<input checked="" type="checkbox"/>	Red	Formation water resistivity at borehole temperature	65	Open
10	IGR	<input checked="" type="checkbox"/>	Green	Linear Index of Gamma Ray	40	Open
11	VSH from GR	<input checked="" type="checkbox"/>	Green	vShale from Gamma Ray	61	Open
12	Final Vshale	<input checked="" type="checkbox"/>	Black	Final Vshale	43	Script
13	RhoM Preliminary	<input checked="" type="checkbox"/>	Brown	Preliminary Matrix Density from reference Well	52	Script
14	Exploratory DPhiT	<input checked="" type="checkbox"/>	Cyan	Exploratory Total Density Porosity	53	Open
15	1 iter RhoM Min Solver	<input checked="" type="checkbox"/>	Brown	First iteration Matrix Density from Mineral Solver	45	Open
16	1 iter Tot Dens Poro ...	<input checked="" type="checkbox"/>	Cyan	First iteration of Total Density Porosity	67	Open
17	1 iter NPhiT	<input checked="" type="checkbox"/>	Cyan	First iteration of NPhiT that includes RhoM correction	60	Open
18	2 iter Tot Dens Poro ...	<input checked="" type="checkbox"/>	Cyan	Second iteration for PhiT that includes NPhiT	78	Script
19	1 iter RhoF	<input checked="" type="checkbox"/>	Orange	First iteration for Fluid Density	96	Script
20	Final RhoF	<input checked="" type="checkbox"/>	Black	Final Fluid Density Interpolated	48	Open
21	2 iter RhoM Min Solver	<input checked="" type="checkbox"/>	Brown	Second iter Matrix Dens from Mineral Solv using va...	92	Open
22	Final RhoM Interpol...	<input checked="" type="checkbox"/>	Black	Final Matrix Density Interpolated	87	Open
23	vClay from Min Solver	<input checked="" type="checkbox"/>	Yellow	Volume of Clay from Mineral Solver	83	Open
24	vSilt from Solver	<input checked="" type="checkbox"/>	Yellow	vSilt from mineral solver	15	Open
25	vQuartz from Solver	<input checked="" type="checkbox"/>	Yellow	vQuartz from mineral solver	98	Open
26	vCalc from Solver	<input checked="" type="checkbox"/>	Yellow	vCalcite from mineral solver	74	Open
27	vDol from Solver	<input checked="" type="checkbox"/>	Yellow	vDolom + vCalcite + vSilt + vClay from mineral solver	14	Open
28	Anhydrite from Solver	<input checked="" type="checkbox"/>	Yellow	vAnh from mineral solver	19	Open
29	NaCl from Solver	<input checked="" type="checkbox"/>	Yellow	vNaCl from mineral solver	54	Open

Runs the selected, checked rows in sequential order, the curve output of a row may be an input to the next row.

The first three rows are user scripts that yield indicator curves for different zones, like formations, members, or other.

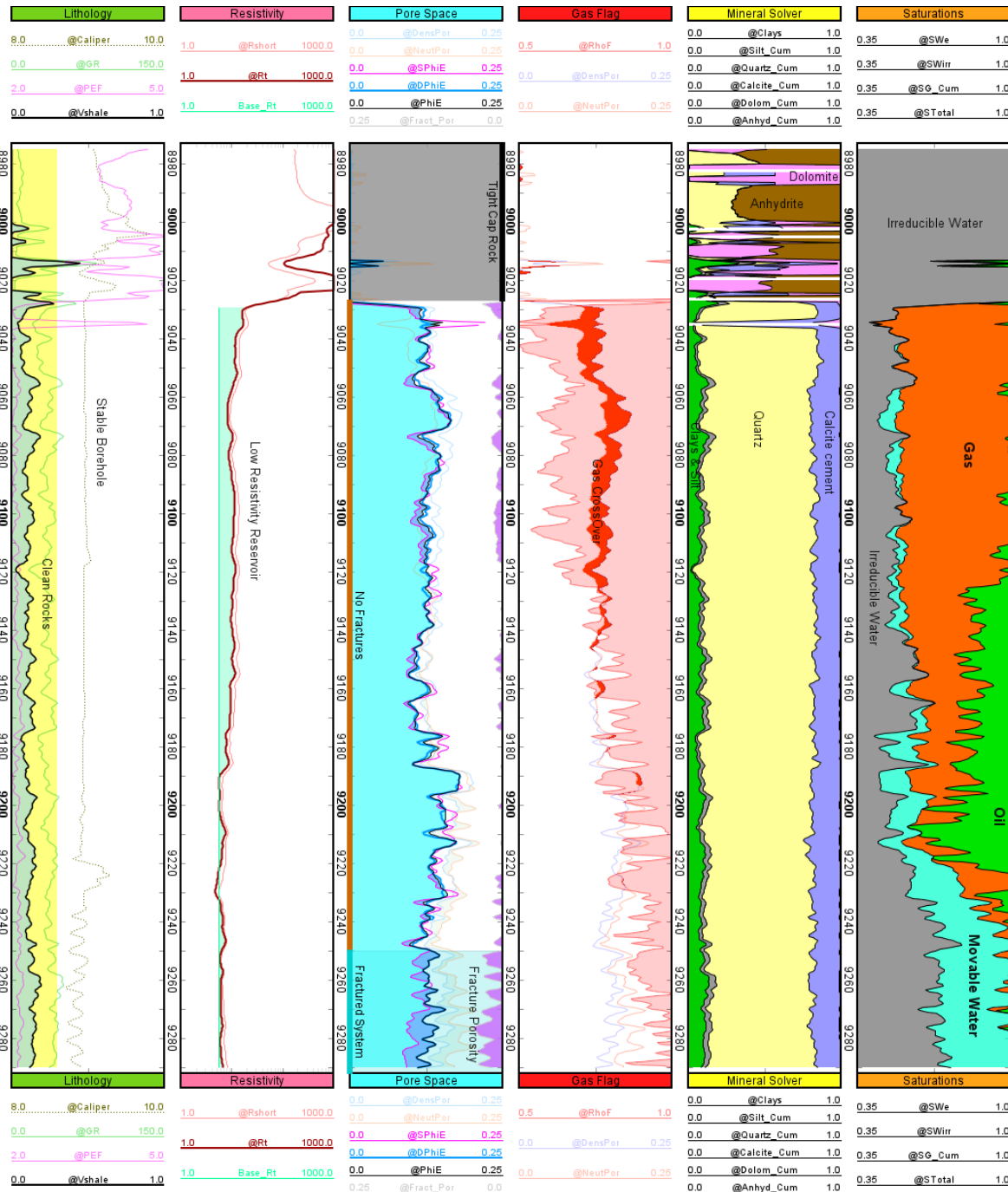
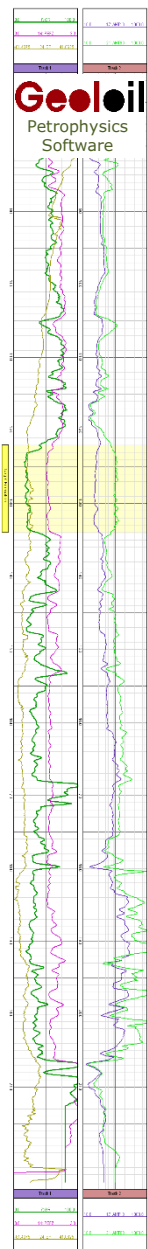
A regular function, just a default built-in algorithm.

This advanced workflow program for a complex reservoir, uses 77 rows combining functions and scripts.

It uses different parameters per formation, so properties like salinity and matrix density are staircase curves on their own, instead of being constant.

The workflow shows computations for the **mineral solver** using pseudo-AI fuzzy sets logic.

**Import** workflow allows to append a workflow of additional rows or load a new one for a new well.  
**Export** workflow exports the checked workflow rows so other wells with the same raw input curves can import it and run it automatically

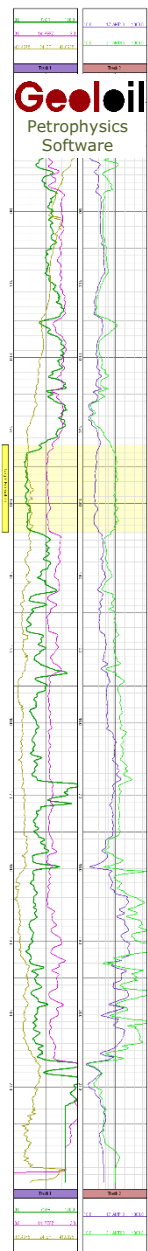


## Workflow output

Running the workflow and displaying the results takes few seconds to some minutes depending on the complexity. Then, the user designed workflow program is applied to other wells to produce immediate results.

This complex workflow outputs:

- Four saturation curves: Irreducible Water Saturation, Movable Water Saturation, Gas Saturation, and Oil Saturation.
- Mineral solver
- Porosity and Fracture Porosity
- Iterated Fluid Density RhoF and its consistency with gas cross-over.



## Each workflow row has its own output panel

GLOG-Example01.glog Folder: C:\Users\oscar\GeolOil\_Marketing\SPWLA\_Dec5\GLOGs

Save as unlock [Icons] LAS PNG C 0.8 1 [Icons]

Functions Workflow Edit function UID=65: Rw

Output Transf. Vshale  $\phi$  Phi RhoM  $\rho$   $\kappa$  Permea.  $\Omega$  Rw Tmp °F  $\Delta$  SW TOC  $\sigma$  Geomec. Miner RhoM  $\rho$  Scripting

SPECS UID 65 = Rw: Formation water resistivity at borehole temperature

Deselect Validate Calculate Calc & Disp Comment: Type a comment for the output curve

Output curve definition:

Mnemonic	Units	UID	Description
Rw	OHMM	FORMC_65	Formation water resistivity at borehole temperatur

Left Outlier Left clip Right clip Right Outlier

☒ Output post-processing: Trim 0.0 10000.0 Accept (K<sub>hor</sub> example)

☐ Comp. only on indic flag @DEPTH 0.0 1000.0 (Depth passes filter when: min<signal<max)

☐ Set const. on indic flag: @DEPTH 0.0 1000.0 0.0 (Example, force  $\phi_e = 0$  if vsh > cutoff) (missed -999.25 allowed)

Left threshold Replacement value: where (curve < Threshold) set (Output = Replacement)

☐ shift left on cutoff: 0.03 0.0 normally set Replacement value < Threshold ( $\phi$  example)

Right threshold Replacement value: where (curve > Threshold) set (Output = Replacement)

☐ shift right on cutoff: 0.85 1.0 normally set Replacement value > Threshold (V<sub>sh</sub> example)

Curve number

☐ Merge result with curve: 0 as background merge over (result will be placed on top of curve number)

Curve Number

Resulting curve stored in: @Rw (The output curve is handled automatically. 0 will append a new curve)

User defined mnemonic output

Post-processing filtering and outlier controls.

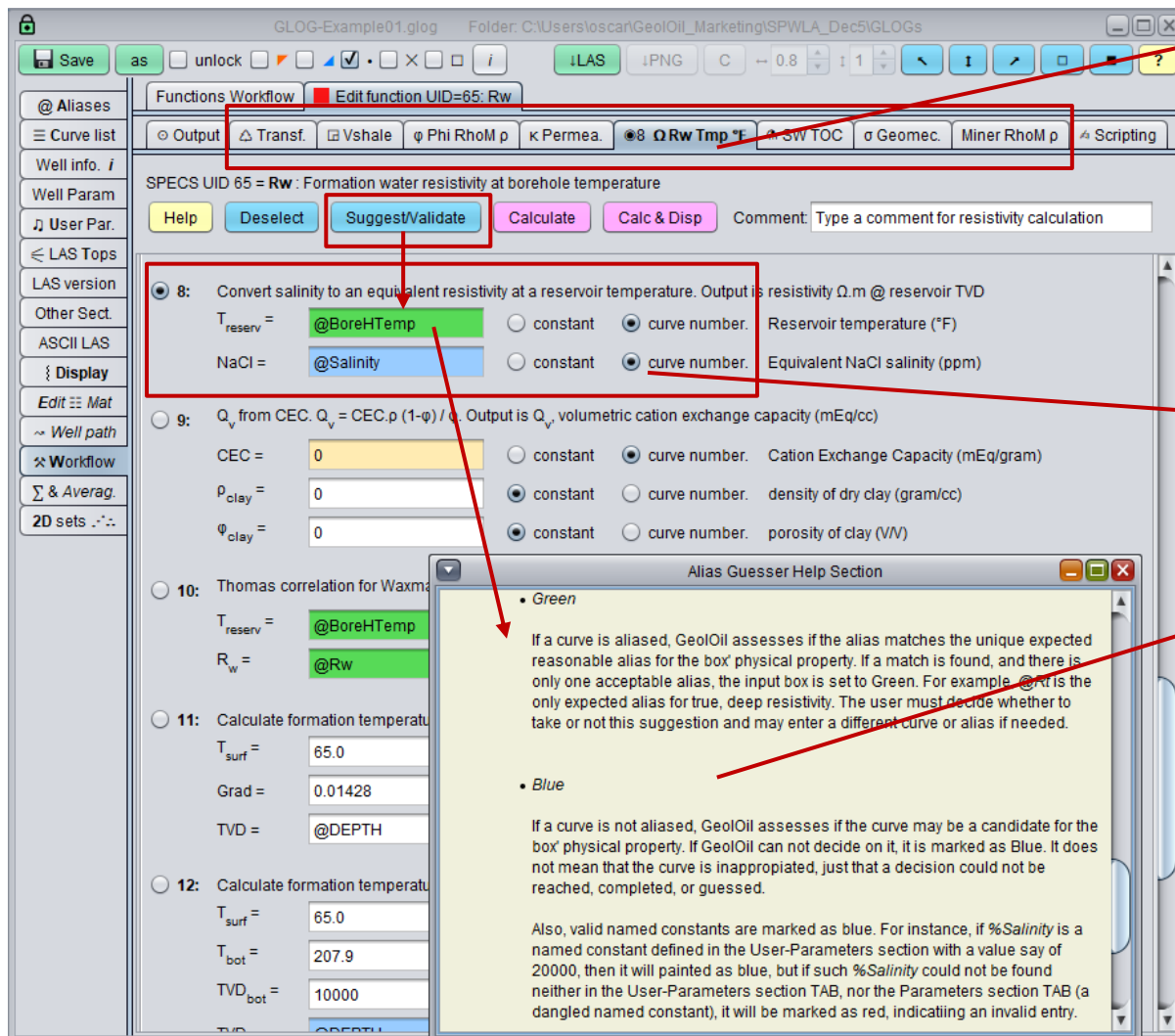
Conditional pre-processing indicator filter.

Post-processing options, including merging over or under former curves.

User defined output alias



## Each workflow function row has its own computations panel



The screenshot shows the 'Functions Workflow' panel in the Geoloil software. The 'Rw' tab is selected, and function 8 is highlighted. A red box highlights the 'Suggest/Validate' button. A red arrow points from the 'Suggest/Validate' button to the 'Alias Guesser Help Section' dialog box.

**Alias Guesser Help Section**

- Green**  
If a curve is aliased, Geoloil assesses if the alias matches the unique expected reasonable alias for the box' physical property. If a match is found, and there is only one acceptable alias, the input box is set to Green. For example, @Rw is the only expected alias for true, deep resistivity. The user must decide whether to take or not this suggestion and may enter a different curve or alias if needed.
- Blue**  
If a curve is not aliased, Geoloil assesses if the curve may be a candidate for the box' physical property. If Geoloil can not decide on it, it is marked as Blue. It does not mean that the curve is inappropriate, just that a decision could not be reached, completed, or guessed.

Also, valid named constants are marked as blue. For instance, if %Salinity is a named constant defined in the User-Parameters section with a value say of 20000, then it will be painted as blue, but if such %Salinity could not be found neither in the User-Parameters section TAB, nor the Parameters section TAB (a dangled named constant), it will be marked as red, indicating an invalid entry.

A large collection of built-in +130 algorithms, from Math Transforms, Vshale, Pore Space, Permeability, Temperature, Saturations & Shale Oil, Geomechanics, and Mineral Solvers.

The eight function on the Rw algorithms tab was selected. This converts Salinity to Rw, using a salinity curve as input, instead of a constant salinity value.

A color code guesses and suggests aliases for each input box. Each input can be a:

- Numerical constant
- Named constant (% Sigil)
- Mnemonic curve
- Aliased curve (@ Sigil)

## Special math function:

*“Safe” range interpolation: Outlier free*

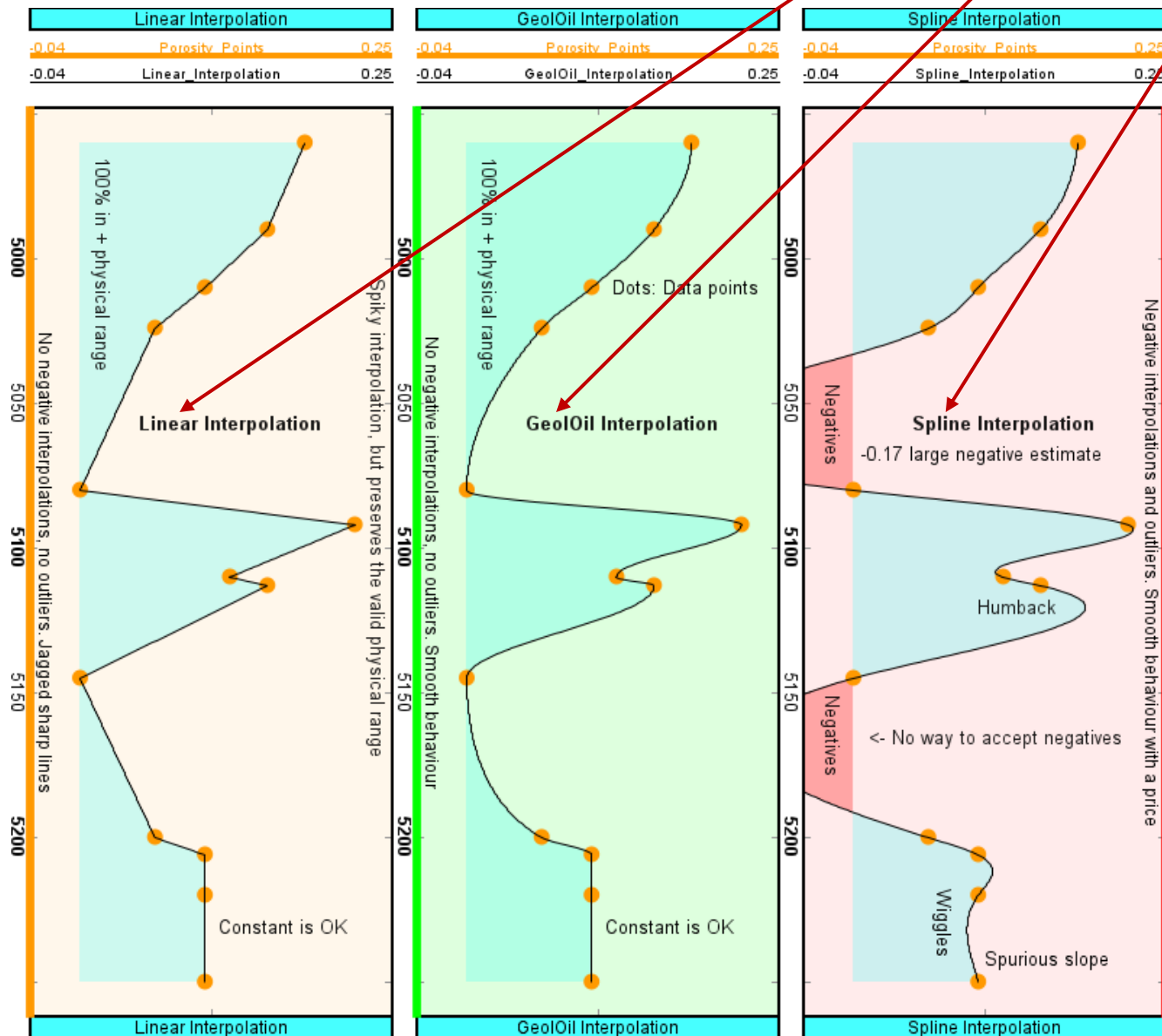
CURVE INTERPOLATION and KERNEL SMOOTHING: Fill void gaps through interpolation, or smooth curves to produce trends.

3: Interpolate and fill -999.25 void gaps in a curve x, up to a maximum depth separation distance

x = PoroPoints Introduce the curve number to interpolate

gap = 1000 Introduce the maximum slot depth aperture to fill with interpolation, 0 to a big number

Curvature 0.50 0: linear interpolation 0.50-0.60 yields good results 1 smoother but may yield outliers





# Special math function: *Uncertainty probability envelopes*

10: Create a moving uncertainty envelope for a curve, like min or max wrapping curve, avg, and quantile confidence bands.

x = @PhiE Introduce the curve number of X to produce a moving envelope probability curve

gap = 100.0 Introduce the maximum slot depth aperture to fill with interpolation, 0 to a big number

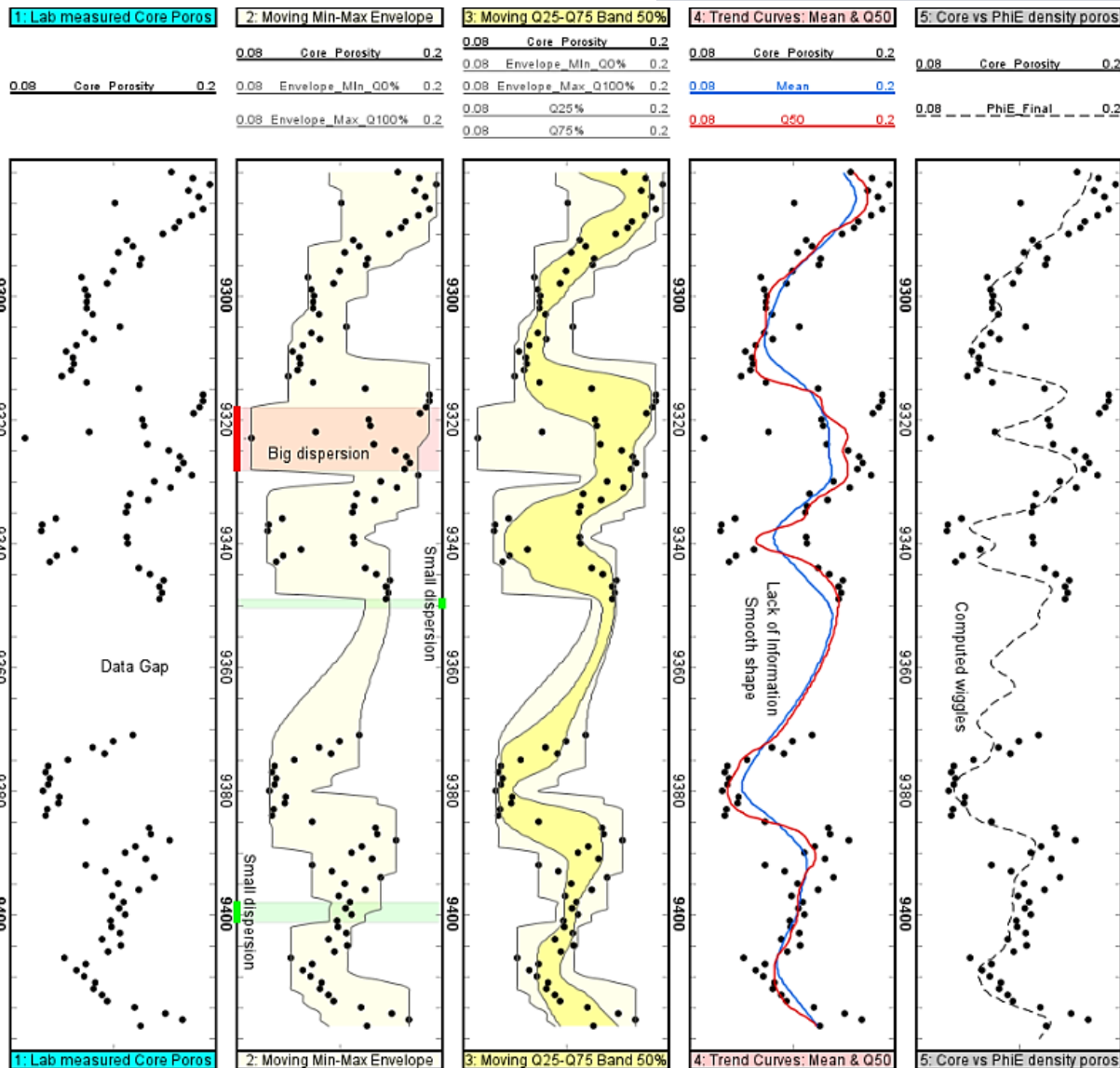
h = 10.0 ☒ constant ☐ curve number, vertical bandwidth to compute moving envelope

☒ If a quantile envelope is selected, specify the left 0.0 ≤ area ≤ 1.0 covered in the h depth band-width

LeftArea = 0.75 ☒ constant ☐ curve number. 0 for min, 0.25 low, 0.50 median, 0.75 high, 1 max

☐ If a generalized moving average is selected, specify its power, like 1.0 for a regular arithmetic mean

Power = 1.0 ☒ constant ☐ curve number. -∞ for min, -1 harmonic, 0 geomet, 1 mean, ∞ max

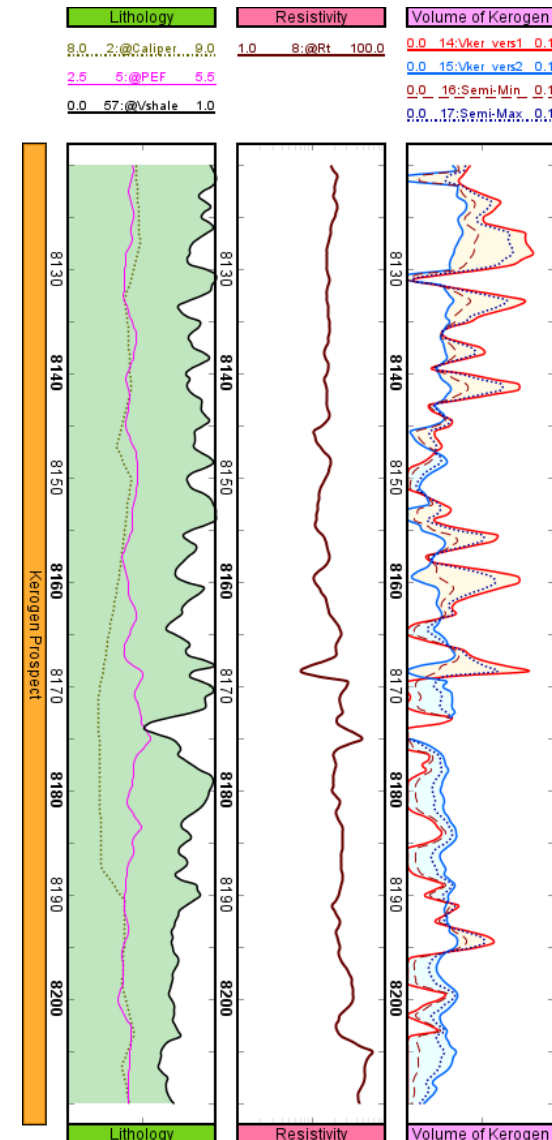
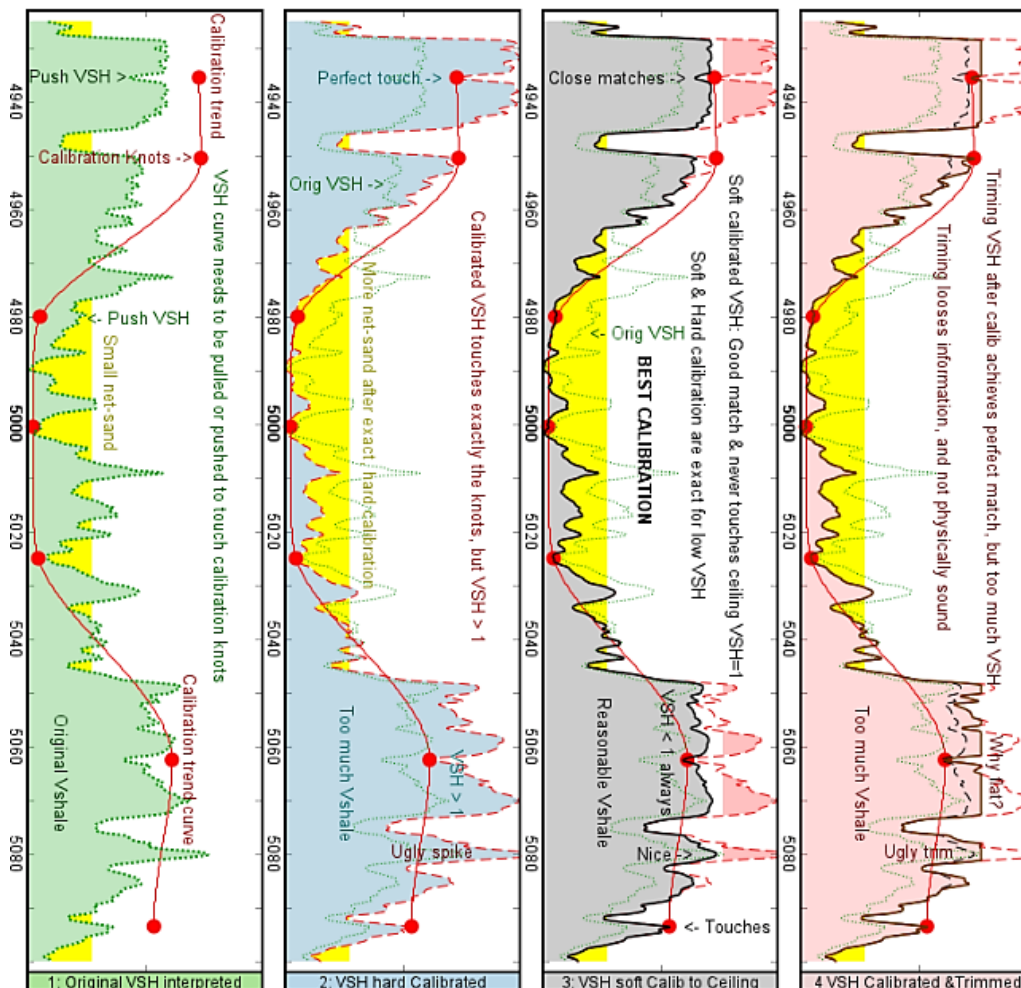


## Special math functions:

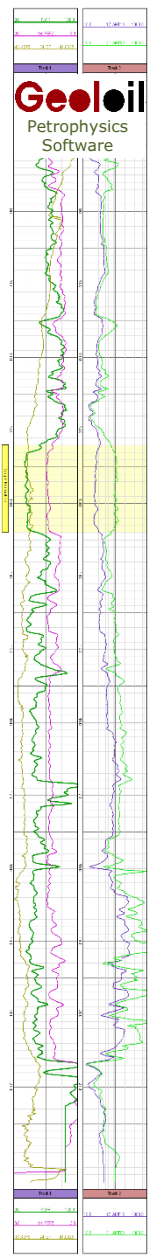
Left: Conditional soft calibration, and

Right: semi-extremes

1: Original VSH Interpreted	2: VSH hard Calibrated	3: VSH soft Calib to Ceiling	4: VSH Calibrated & Trimmed
0.0 2:VSH_CalibNodes 1.3	0.0 2:VSH_CalibNodes 1.3	0.0 2:VSH_CalibNodes 1.3	0.0 2:VSH_CalibNodes 1.3
0.0 3:VSH_InterpTrend 1.3	0.0 3:VSH_InterpTrend 1.3	0.0 3:VSH_InterpTrend 1.3	0.0 3:VSH_InterpTrend 1.3
0.0 4:Orig_VSH 1.3	0.0 4:Orig_VSH 1.3	0.0 4:Orig_VSH 1.3	0.0 4:Orig_VSH 1.3
0.0 5:VSH_Calib 1.3	0.0 5:VSH_Calib 1.3	0.0 5:VSH_Calib 1.3	0.0 5:VSH_Calib 1.3
0.0 6:VSH_Ceiled 1.3	0.0 6:VSH_Ceiled 1.3	0.0 6:VSH_Ceiled 1.3	0.0 6:VSH_Ceiled 1.3
0.0 7:VSH_Calib_tin 1.3	0.0 7:VSH_Calib_tin 1.3	0.0 7:VSH_Calib_tin 1.3	0.0 7:VSH_Calib_tin 1.3







## The Scripting engine takes curve aliases and named constants

Just four lines of source code. Try to do this in Python. This is a dedicated, domain specific scripting language for petrophysics and log processing.

Geoloil Petrophysics

File Save GLOGS Lock/Unlock LockMode: Network Rock Properties

masterWebWorkflow.glog Folder: D:\Projects

Aliases Curves Well info. Parameters LAS version LAS Tops Other Sect. LAS text Display Mat Edit Well path Workflow & Filters 2D sets

Functions Workflow Edit function UID=46: Final Effective Poros

Export Function Deselect Calc

SPECS UID 46 = Final Effective Poros: Final Effective Porosity from Density and Neutron

User stream script:

```
# - FUNCTIONS: abs(), exp(), ln(), log10(), sin(), cos(), tan(), asin(), acos(), atan(), isValid(), valueOrZero()
# Indicator func: ind(x,y) = 1 if x < y, 0 if x >= y, -999.25 if decision fails as x,y could be -999.25
# func: ind(x,y,z) = 1 if (x<y) and (y<z), 0 if at least one condition is false, -999.25 unknown.
# To force an empty result (set it to -999.25) use the function blank(x,y) or blank(x,y,z) as pre-multiplier
# blank(x,y) = -999.25 if if (x<y), 1 if x >= y, or decision fails as x,y could be -999.25
# blank(x,y,z) = -999.25 if if (x<y) and (y<z), 1 if at least one condition is false or unknown
# blankOutside(x,y,z) = -999.25 if y does not belong to interval x<=y<=z, 1 if y is inside interval
# trim (left,x,right) forces x in interval, otherwise = left if x < left, or = right if x > right
# shiftCurve(x,n) vertically shifts curve x downwards n depth steps if n>0, or upwards n depth steps if n<0
# shiftLeft (x,cutoff,value) horizontally shifts x: where (x < cutoff) set (x = value). shiftRight() uses >
# min(x,y) returns the smaller of x and y, or -999.25 if either x or y are unknown. max(x,y) returns larger
# merge (x,y) = x if x is known; = y, if x is unknown or -999.25 (transparency): merge puts x on top of y
# avg (x1,x2) returns average skipping -999.25 values. min(), max(), merge(), avg() can use 3 or more curves
# - USER DEF FUNC Example: def average(x,y) (return((valueOrZero(x)+valueOrZero(y)) / (isValid(x)+isValid(y))))
# ===== You may delete the Help above. Please script your code below =====
```

Reg\_DPhiE = @43  
Safe\_PhiE = @45

weight = 0.90

Final\_PhiE = merge( (1-weight)'Reg\_DPhiE + weight'Safe\_PhiE, Safe\_PhiE, Reg\_DPhiE)

Output messages: Powerful built-in GLS multi-argument merge() function: If the first argument is not available (-999.25), it will look for the second argument, if the second argument is not available, it will look for the next...

Curve list: \* is a Display curve. Click [?] for help

1	@DEPTH	29	@RHOM_Contrast /F40
2	* @Caliper	30	@RHOM_Solver /F14
3	CT90	31	@RhoM /F54
4	DPHI	32	* @DensPor /F32
5	@DensCorr	33	@NeutPor /F52
6	* @GR	34	@vShale_Neut /F33
7	* @NPorLim	35	* @vShale /F20
8	* @PEF	36	* @Clays /F30
9	@RhoB	37	* @Silt_Cum /F82
10	* @Rt	38	* @Quartz_Cum /F75
11	@Rshort	39	* @Calcite_Cum /F12
12	RT20	40	* @Dolom_Cum /F66
13	@Rmedium	41	@Phi_Rxo /F60
14	RT60	42	NPhiE /F42
15	RT90	43	@PhiE_Dens /F70
16	* @Rxo	44	@Rugosity_Flag /F79
17	* @SP	45	@PhiE_Robust /F16
18	TENS	46	* @PhiE /F46
19	QF	47	* @KhorMax /F15
20	QN	48	@SWirr_Prel /F38
21	@NaCl_Form_A /U13	49	@SWe_Prel /F63
22	@NaCl_Form_B /U96	50	* @SWe /F59
23	@BoreHTemp /F68	51	* @SWirr /F89
24	@Salinity /F41	52	* @BVW /F91
25	@Rw /F19	53	* @BVWirr /F90
26	@Rmf /F64	54	@Rwa /F85
27	@IGR /F29	55	NaCl_app /F74
28	@vShale_GR /F95		

4 lines of source code

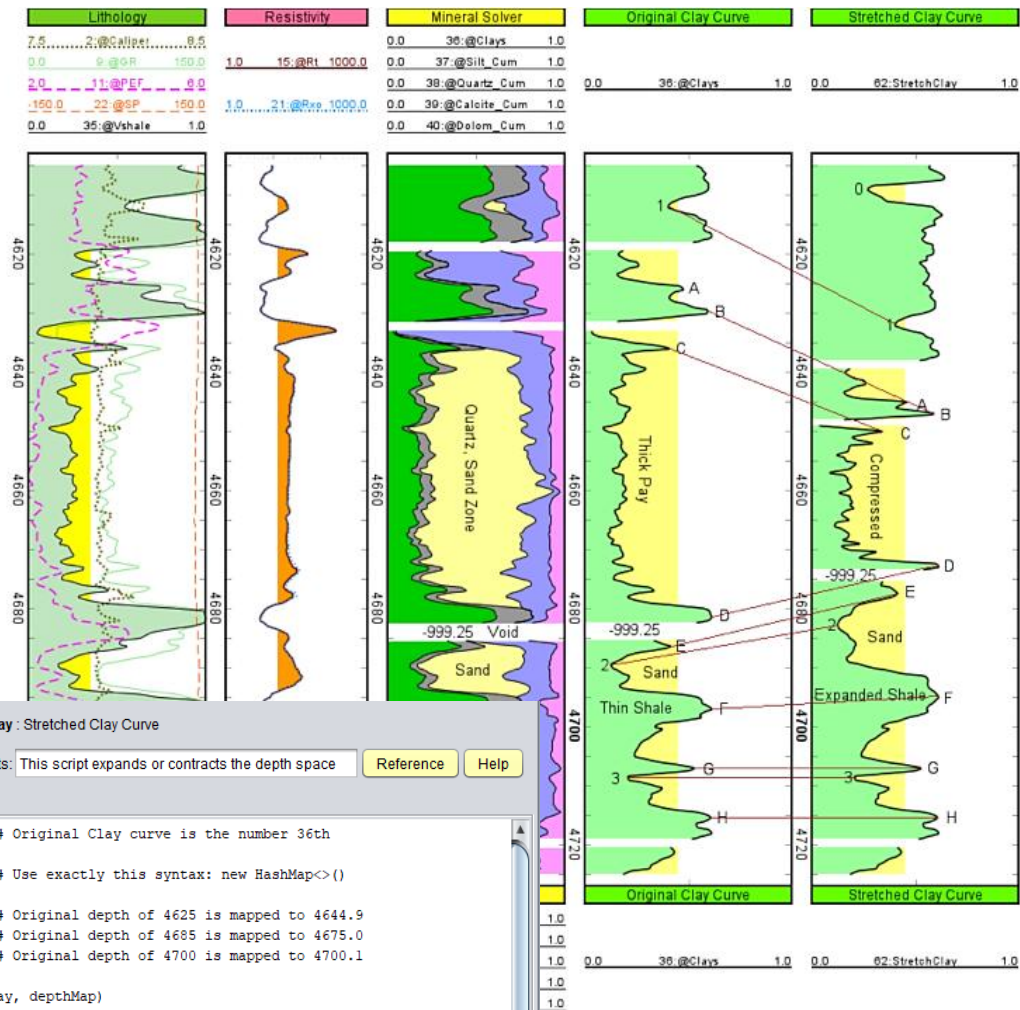
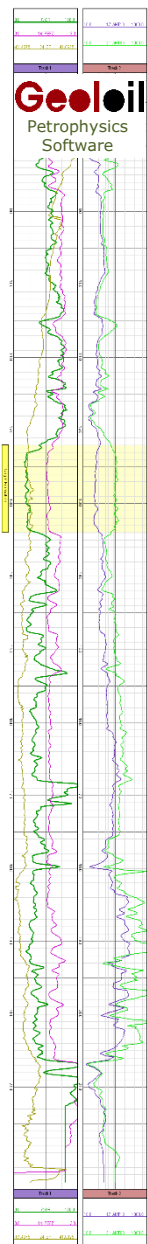
Both curve numbers 43 and 45 are aliased as @PhiE\_Dens and @PhiE\_Robust. Any other wells using these aliases will found its MultiWell equivalence

Any Script or regular function can be exported or included into a work-flow. Make sure that curves are aliased to allow Multi-Well curve equivalence in other wells.

How to mouse on the ? question mark to see the list of curves: \* indicates that the curve is plotted @ indicates that the curve is aliased

# A Script example to stretch a log curve

Just six lines of source code. (return command is optional)



ExportFunction

SPECS UID 25 - Stretched Clay : Stretched Clay Curve

Deselect

Calculate

☒ Active

Comments: This script expands or contracts the depth space

Reference

Help

User Script:

```

1 Clay = @36 # Original Clay curve is the number 36th
2
3 def depthMap = new HashMap <> (); # Use exactly this syntax: new HashMap<>()
4
5 depthMap.put (4625.0, 4644.9); # Original depth of 4625 is mapped to 4644.9
6 depthMap.put (4685.0, 4675.0); # Original depth of 4685 is mapped to 4675.0
7 depthMap.put (4700.0, 4700.1); # Original depth of 4700 is mapped to 4700.1
8
9 def deformedDepthClay = stretch (Clay, depthMap)
10
11 return (deformedDepthClay) # Return creates a new stretched curve from Clay
12
13 # That is all: Line 3 declares a depth map correspondence Original - Destiny
14 # Lines 5-6 defines a contraction of depths, like a pinchout thinning
15 # Lines 6-7 defines an expansion depths, like stretching a rubber band
16 # Line 9 stretch() performs the contraction-expansion, defining a new curve
17 # Line 11 returns and exits from the script.
            
```

Output messages, warnings and errors:

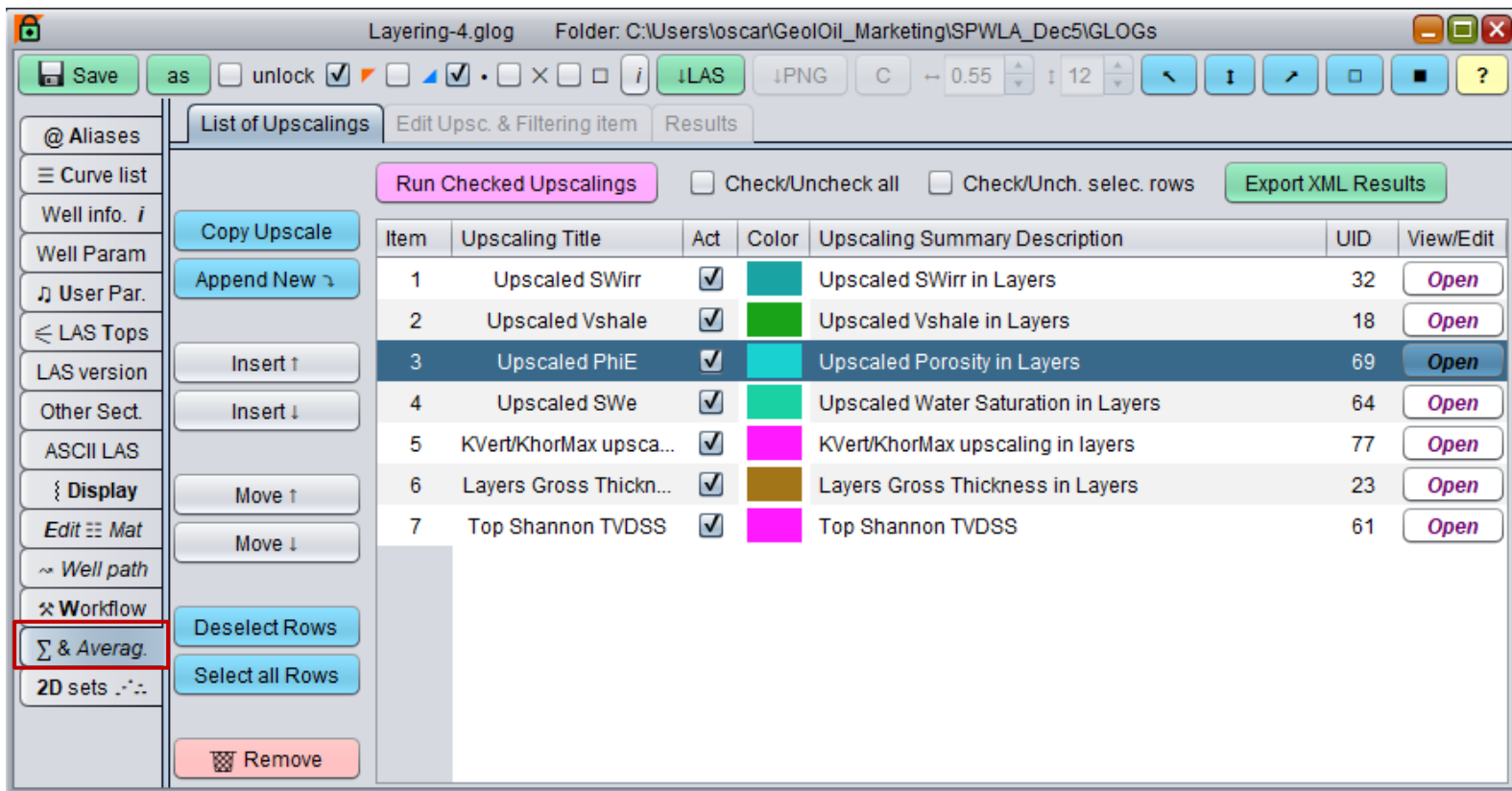
\*\* SUCCESSFUL Script compilation and execution. \*\*

Please validate the result with 'Edit Table' or by Plotting the curve.

## The Upscalings Workflow

Similarly to the Function & Scripting workflow, a WorkFlow of Upscalings and Petrophysical Summaries framework is defined.

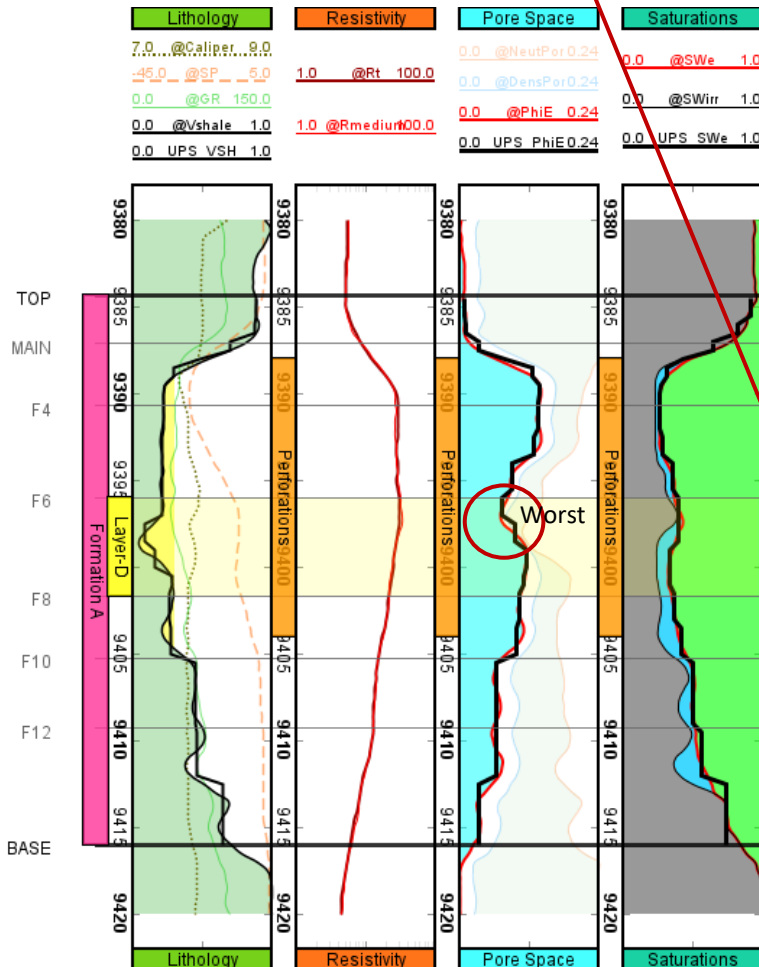
This allows (among other things) to define a layering strategy to build 3D geo-cellular models, compute petrophysical cutoffs, net-rock and net-pay per zones, and other computations.





## A layering framework for the zone "Layer-D"

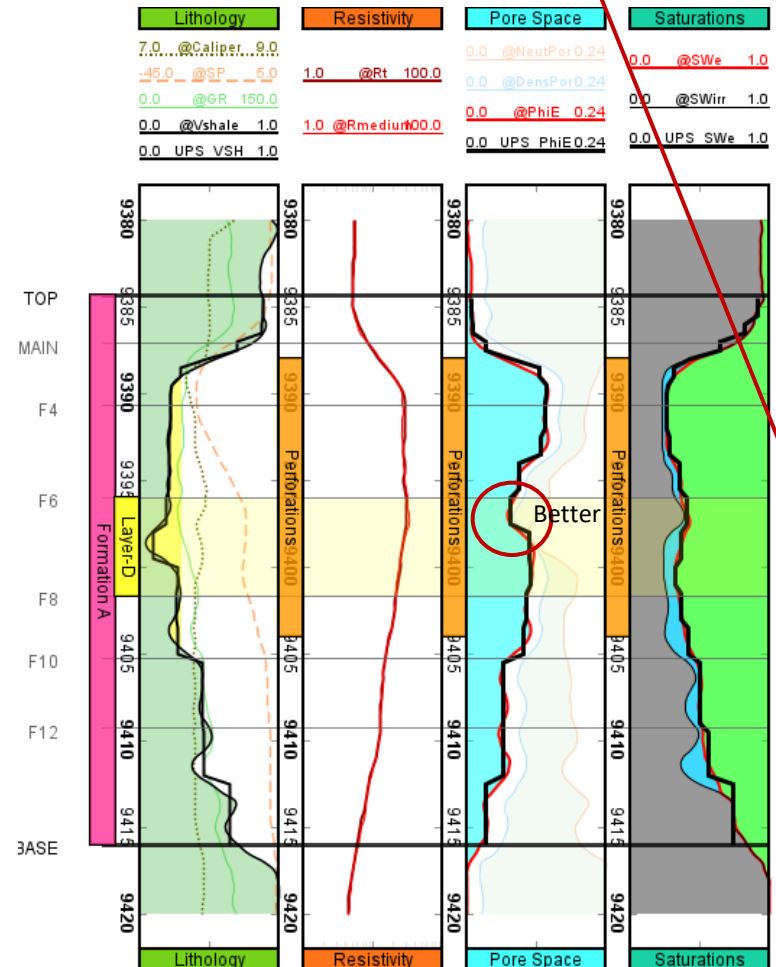
### 4 layers partition



Reservoir Simulation Cells



### 3 layers partition

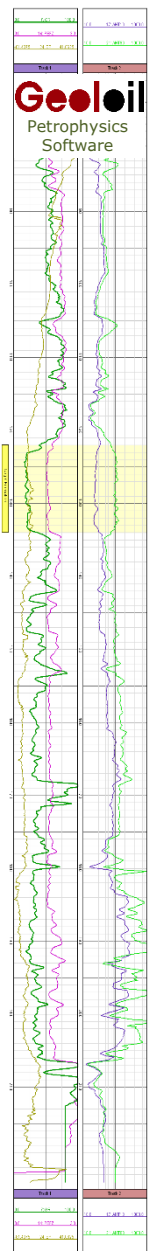


Reservoir Simulation Cells



Not always a finer vertical resolution yields a better model to capture reservoir heterogeneity and baffles. The split of "Layer-D" to the right into only 3 layers follows better the porosity than the partition into 4 layers to the left. This visual representation of the upscalings, is essential to define to good 3D geo-cellular models for simulation purposes.





# Thank you!




"A great petrophysics software must excel in four aspects:  
 1.- Designed after decades of petrophysics consulting.  
 2.- Modern programming practices in compiler design and threads.  
 3.- Advanced & innovative applied mathematical algorithms.  
 4.- An outstanding and fast customer service.  
 Well, we have all these traits." **Oscar Gonzalez**, GeolOil LLC, Director ■

## TESTIMONIALS↓

48 testimonials→

26 Google reviews→

**2025 November:** "I continue to transition from PowerLog to GeolOil, and I am now recommending GeolOil to companies for which I consult. Congratulations on a fine and ingenious product. Cheers."

 **Douglass Sharp**. Independent Senior PetroPhysicist Consultant. Texas, USA. ■


**2024 August:** "I think GeolOil is amazing. I've been using the software on almost a daily basis and the petrophysical capabilities are making my life so much easier. "

 **Emre Cankut Kondakei**. Senior Geologist. IPT Well Solutions. Houston, Texas, USA. ■

**2024 July:** "The GeolOil software is working nicely on a Macintosh computer. I have to admit that initially I was too dummy to get GeolOil started on Mac-OS, but a colleague helped me to get on a right track."

 **Heikki Bauert**. Geological Survey of Estonia (EGT), an Estonian Republic Government Agency. Geological Survey of Estonia. Rakvere, Estonia. ■


**2024 January:** "... In a world where exceptional service is increasingly rare, you have truly set a standard to aspire to, and I wanted to take a moment to acknowledge and appreciate your outstanding contribution. Your dedication makes a real difference, and we are grateful to have such a reliable partner." [Read more](#).

 **Enis Aliko**. Senior Drilling Engineer. Wellynx Engineering. Pescara, Italy. ■

**2023 August:** "I've been impressed by the programme, and it works for my purposes on my small laptop."

 **Michael McCaughey**. Senior GeoScientist, ELGOL Geoscience. Director. Twyford, England, UK. ■

**2023 February:** "I was a Geolog user for many years so I know it pretty well. GeolOil is really good, it just takes a while to get used to like any software."

 **Sheldon Murphy**. Senior Petrophysics Consultant. Epoch Geologic, LLC. Pittsburgh, Pennsylvania, USA. ■