SasView
Coding Beyond the Models

Jeff Krzywon, NIST Center for Neutron Research
June 15, 2022

Outline

- Getting started
- Python and dependencies
- SasView architecture and code structure
- ‘Entry points’
- Advanced topics

Large Learning Curve

- Massive code base
  - Sasview 100,000+ lines of python and c
  - sasmodels, bumps, periodictable, etc.
- Multiple non source code directories
- Multiple branches
  - main branch (v5.x) only recently became the default (2021)
  - Development and release branches
- Side project for most contributors
  - Occasional student (temporary and with a focused project)
  - Others ‘when they can find the time’

[1] https://github.com/SasView/sasview/wiki/DevNotes_DevGuide_GettingStarted

Getting setup

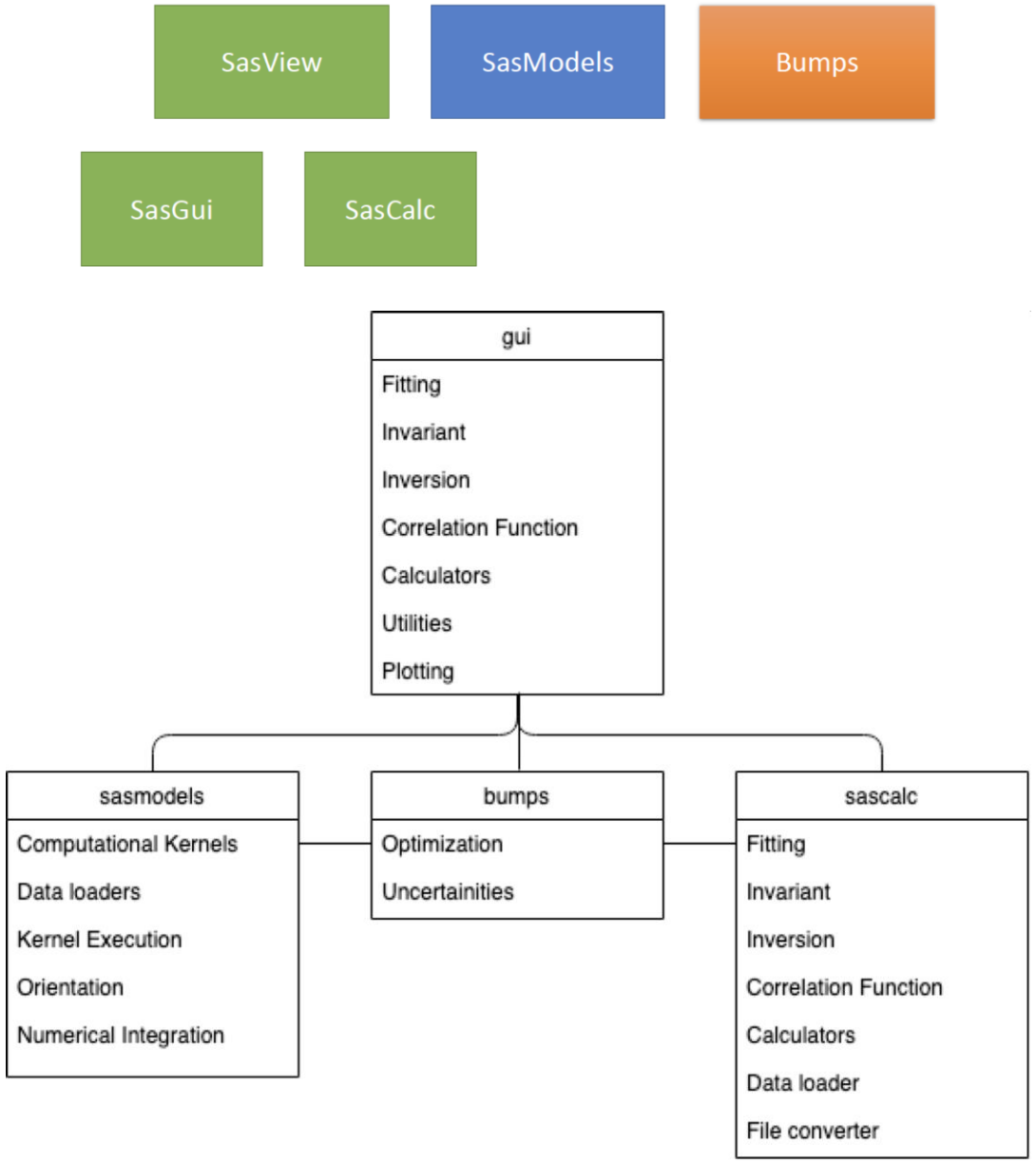
- Getting started wiki[1]
  - Developer environment setup
    - WIP -> requirements.txt
  - Code overview and guidelines
- Environments are important!
  - Base python installation (v3.9)
    - Built in packages
      - os, sys, math, logging, etc.
    - External packages as needed
      - pip install <package_name>
      - Built on top of base installation!
    - Environments create separate pathway for packages
    - Create and activate new environment
      - conda
        - conda create --name <myenv>
        - conda activate <myenv>
      - Python
        - python3 -m venv <venv>
        - C:\> <venv>\Scripts\activate.bat

- [1] https://github.com/SasView/sasview/wiki/DevNotes_DevEnviroment

Python Package Dependencies

PyQt GUI	Matplotlib Plotting utility
Numpy and Scipy Array, numerical, and scientific calculations	Lxml, Hdf5 and h5py Load and Save NeXus and XML files
Twisted, cuda, pyopencl & numba Threading and GPU utilization	Sphinx Documentation
Tinycc C compiler	Periodictable Atomistic data

SasView Architecture



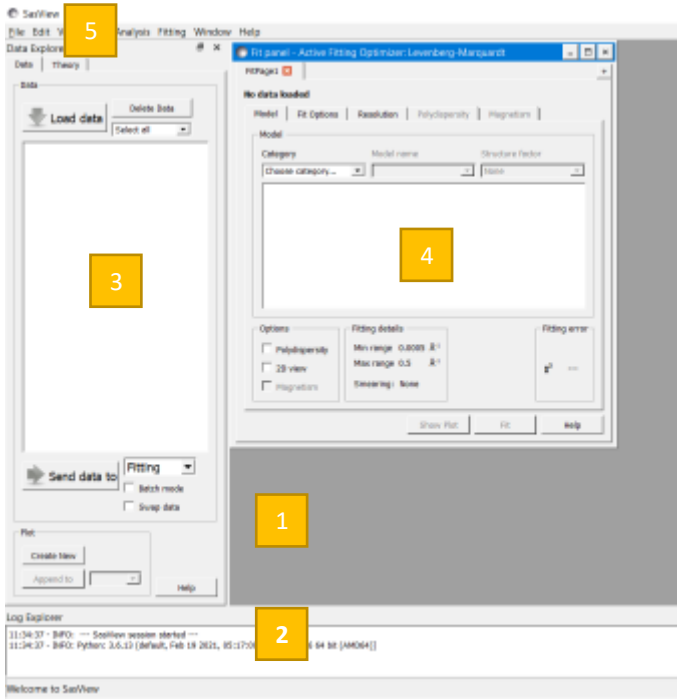
SasView Code Structure

- sasview
  - /.github/workflows – Github actions
  - /build_tools – requirements.txt
  - /docs – Documentation and tutorials
  - /src/sas – Main source directory
    - /qtgui – GUI elements for v5.0+
      - /<subPackage>
      - /UI – Qt and python files
      - /UnitTesting – Unit tests specific to subPackage
      - <filename>.py
    - /sascalc – Calculation elements
    - /sasview – Main window loader for pre v5.0, example data
  - /test – Unit tests for sascalc
- sasmodels
  - /.github/workflows – Github actions
  - /doc – Documentation
  - /example – Example workflows and data
  - /sasmodels – Main source directory
    - kernal(cl|cuda|dll|py).py – kernel model wrappers
    - Mixture.py – Combine models (A*B + C*D)
    - resolution(2D).py – resolution calculators
    - Product.py – P(Q)*S(Q)
    - Sesans.py – sesans transformations
    - Weights.py – Dispersion generators (gaussian, log normal, rectangular, etc.)
    - /models – location of all built-in models

package/within/qtgui

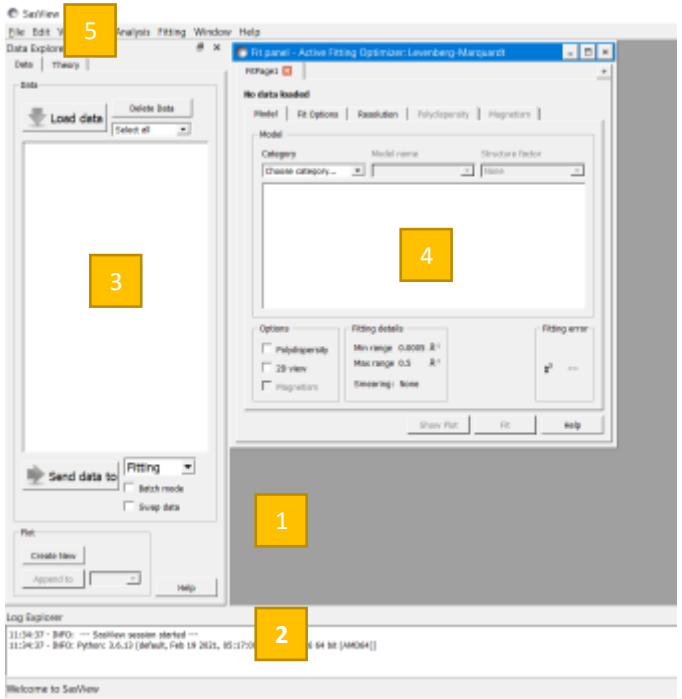
Guide to reading entry point charts

No.	qtgui	sascalc	Description
Label Number	VariableName: Variable name and location associated with labelled GUI element actionName: Method triggered when specific GUI element action is triggered	Path.to.method_called() Path.to.sub.method_called() All paths relative to src.sas.sascalc unless other specified	Text describing the GUI element and purpose within SasView



Entry Points: Main Window

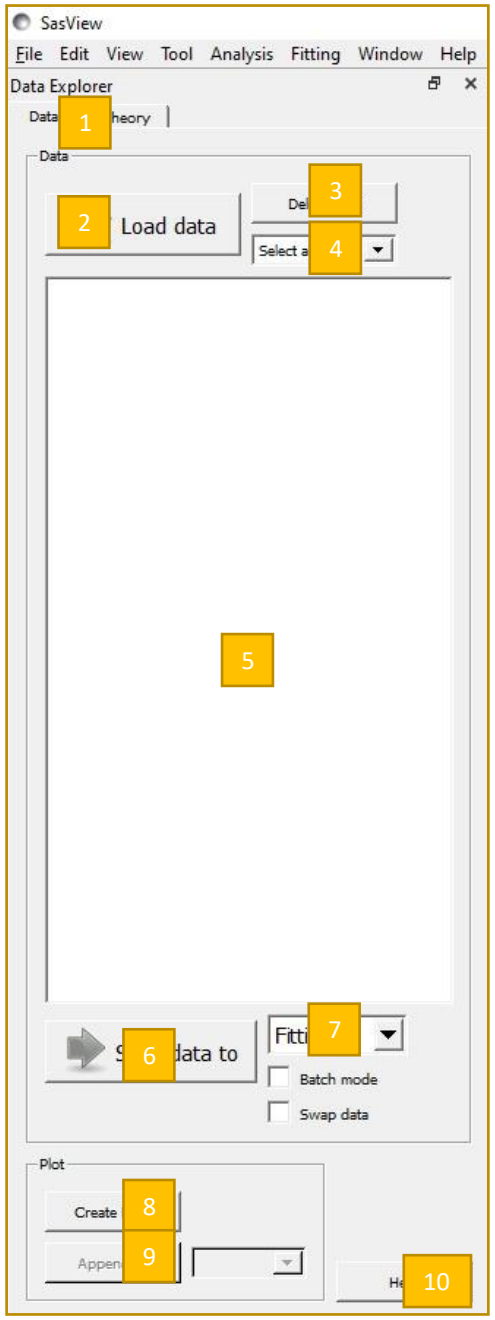
No.	qtgui	sascalc	Description
1	Qtgui.MainWindow.MainWindow.py Qtgui.MainWindow.GuiManager.py	-	Main SasView Window
2	Qtgui.Utilities.SasviewLogging.py Qtgui.GuiManager -> listWidget	Src.logger_config.py Src.logging.ini	Console log
3	qtgui.MainWindow.DataExplorer.py Qtgui.MainWindow.DataManager.py Qtgui.MainWindow.DoppableDtaLoadWidget.py	Sascalc.dataloader Sascalc.dataloader.readers	Data explorer
4	Qtgui.perspectives.Fitting Qtgui.Perspectives.Corfunc Qtgui.Perspectives.Invariant Qtgui.Perspectives.Inversion	Sascalc.fit Sascalc.corfunc Sascalc.invariant Sascalc.pr	Perspectives (default: Fitting)
5	Qtgui.MainWindow.UI.MainWindowUI.py Perspectives: Qtgui.Mainwindow.GuiManager.py	-	Drop down menu bar



Entry Points: Data Explorer

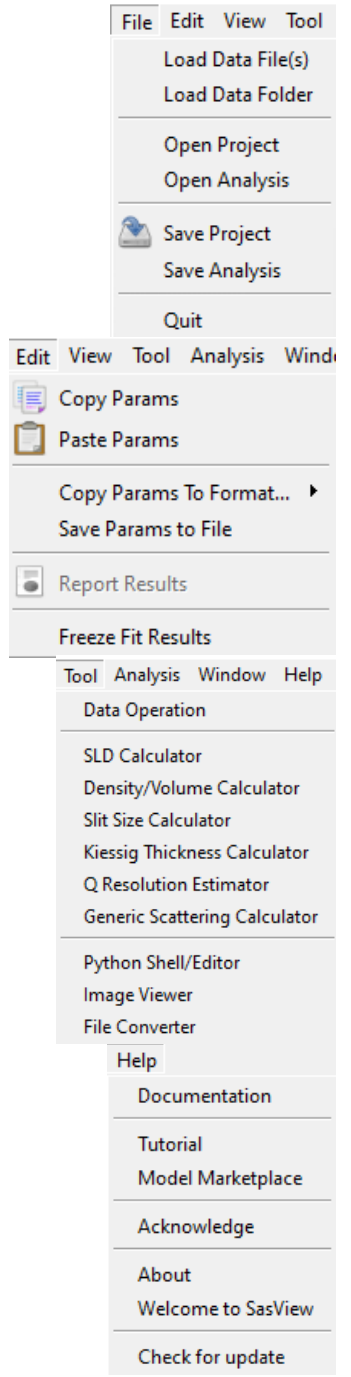
Qtgui/MainWindow/DataExplorer

No.	qtgui	sascalc	Description
1	Variables: DataExplorer.treeView & DataExplorer.freezeView currentChanged: MainWindow.DataExplorer.changeTabs()	-	Data and Theory Tabs
2	Variable: DataExplorer.cmdLoad Clicked: MainWindow.DataExplorer.loadFile() (Menu Option): MainWindow.DataExplorer.loadFolder()	dataloader.loader.load() Dataloader.file_reader_base_class.read()	Open a file browser and load selected files
3	Variable: DataExplorer.cmdDeleteData Clicked: MainWindow.DataExplorer.deleteFile() -> [<Perspective>.removeData()]	-	Delete data that are checked and remove data from all perspectives
4	Variable: DataExplorer.cbSelect Activated: MainWindow.DataExplorer.selectData()	-	Check or uncheck highlighted data sets
5	Variable: DataExplorer.model itemChanged: MainWindow.DataExplorer.onFileListChanged()	-	List of data
6	Variable: DataExplorer.cmdSendTo Clicked: MainWindow.DataExplorer.sendData()	-	Send selected data set(s) to the perspective selected in [7]
7	Variable: DataExplorer.cbFitting Clicked: MainWindow.DataExplorer.updatePerspectiveCombo() -> MainWindow.GuiManager.perspectiveChanged()	-	Changes the active perspective
8	Variable: DataExplorer.cmdNew Clicked: MainWindow.DataExplorer.newPlot()	-	Create new plots for the selected data
9	Variable: DataExplorer.cmdAppend Clicked: MainWindow.DataExplorer.appendPlot()	-	Append selected data to existing plot
10	Variable: DataExplorer.cmdHelp Clicked: MainWindow.DataExplorer.displayHelp()	-	Open data help window



Entry Points: Main Menu Options

No.	Menu Option	qtgui	sascalc	Description
File	Load Data File(s)	MainWindow.DataExplorer.loadFile()	dataloader.loader.load()	Load a list of data files
	Load Data Folder(s)	MainWindow.DataExplorer.loadFolder()	dataloader.loader.load()	Load all data within a specific folder
	Open Project	MainWindow.GuiManager.actionOpen_Project()	-	Open a saved project
	Open Analysis	MainWindow.GuiManager.actionOpen_Analysis()	-	Open a saved analysis
	Save Project	MainWindow.GuiManager.actionSave_Project()	-	Save the existing project
	Save Analysis	MainWindow.GuiManager.actionSave_Analysis()	-	Save the existing analysis
	Quit	MainWindow.GuiManager.actionQuit()	-	Close the program
Edit	Copy Params	GuiManager. actionCopy()	-	Copy the fit parameters from a fit page
	Paste Params	GuiManager.actionPaste()	-	Paste copied parameters into the fit page
	Copy Params To Format…	GuiManager.actionExcel() -or- actionLatex()	-	Copy parameters to Excel or Latex format
	Save Params To File	GuiManager.actionSaveParamsAs()	-	Export parameters to file
	Report Results	GuiManager.actionReport()	-	Create fit report
	Freeze Fit Results	GuiManager. actionFreeze_Theory()	-	Freeze a theory to view later
Tool	Data Operation	Calculators.DataOperationUtilityPanel	-	Add/sub/multi/div two data sets
	SLD	Calculators.SldPanel	Periodictable.xsf/nsf	SLD Calculator
	Density/Volume	DataOperations.DensityPanel	Periodictable	Density/Volume Calculator
	Slit Size	DataOperations.SlitSizePanel	calculator.slit_length_calculator	Slit Length Calculator
	Kiessig Thickness	DataOperations.KiessigPanel	calculator.kiessig_calculator	Kiessig Thickness Calculator
	Q Resolution	Calculators.ResolutionCalculatorPanel	calculator.resolution_calculator	Resolution Calculator
	Generic Scattering	Calculators.GenericScatteringPanel	calculator.sas_gen	Real Space Calculator
	Python Shell/Editor	Utilities.IPythonWidget	-	Python Editor Window
	Image Viewer	Utilities.ImageViewer	dataloader.manipulations	Convert images into Arb Int Data
	File Converter	Utilities.FileConverter	file_converter. FileConverterUtilities	Convert non-standard files
Help	Documentation	GuiManager.actionDocumentation()	-	Open a browser with documentation
	Tutorial	GuiManager.actionTutorial()	-	Open the latest toturial
	Model Marketplace	GuiManager.actionMarketplace()	-	Open a browser to the Model Marketplace
	Acknowledge	GuiManager.actionAcknowledge()	-	How to Acknowledge SasView
	About	GuiManager.actionAbout()	-	About Sasview (ver, contributors, etc.)
	Welcome to SasView	GuiManager. actionWelcome()	-	Splash screen
	Check for update	GuiManager.actionCheck_for_update()	-	Check your version against the latest version

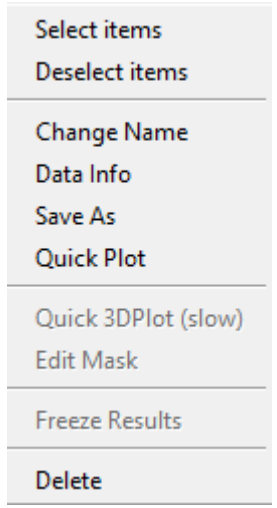


Qtgui/MainWindow/DataExplorer

Entry Points: Data Context Menu

Sascalc/dataloader

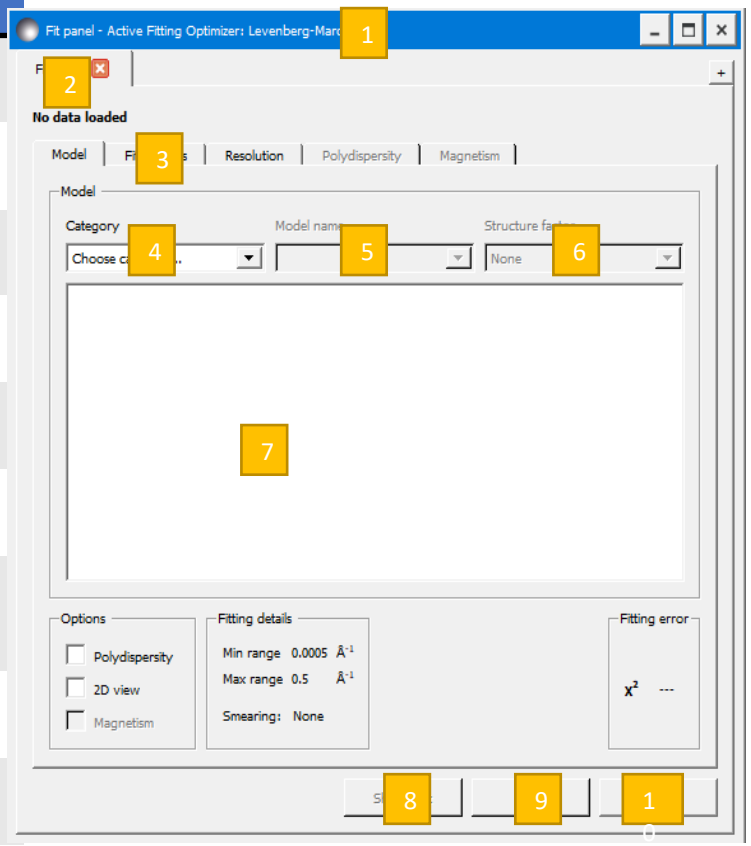
	qtgui	sascalc	Description
Select items	onFileListSelected()	-	Check the boxes of the highlighted data (Ctrl/Shift/Cmd)
Deselect items	onFileListDeselected()	-	Uncheck the boxes of the highlighted data sets (Ctrl/Shift/Cmd)
Change Name	changeName() -> nameChangeBox.show()	-	Open a dialogue to change the display name of the data set
Data Info	showDataInfo() -> txt_widget.show()	Data_info.Data1D.__str__() Data_info.Data2D.__str__()	Show the extended data info including meta data
Save As	saveDataAs()	Loader.save()	Save the selected data set as multi-column text, CVS, canSAS XMl, NXcanSAS
Quick Plot	quickDataPlot()	-	Create a quick plot of the data
Quick 3D Plot	quickData3DPlot()	-	Create a one-off heat-map style plot from a 2D data set
Edit Mask	showEditDataMask()	-	Mask selected data points for 2D data
Freeze Results	freezeSelectedItems()	-	Freeze the theory and send it to the theory tab
Delete	deleteSelectedItem()	-	Delete the data set and remove it from all perspectives



Entry Points: Fitting Perspective

src/sas/qtgui/Perspectives/Fitting
Src/sas/sascalc/fit
sasmodels

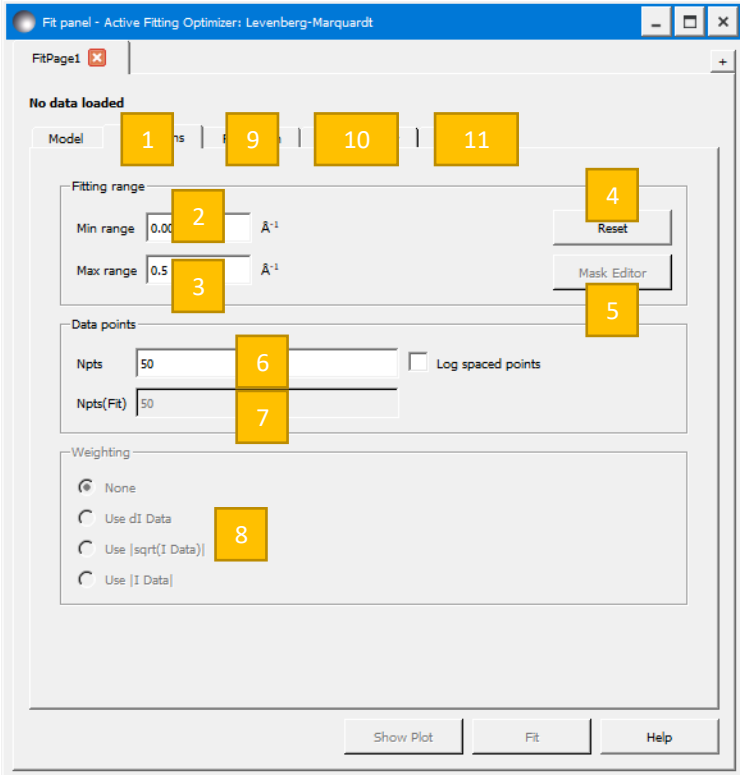
No.	qtgui	sascalc	Description
1	FittingPerspective.__init__()	-	Fit perspective – generated on SasView open
2	FittingWidget.__init__() & FittingLogic.__init__()	-	Fit tab – generate on loading SasView or when new data sent to fitting perspective
3	OptionsWidget.__init__() -> FittingOptions.__init__()	-	Fitting Options – generated with new fit tab
4	Variable: FittingWidget.cbCategory CurrentIndexChanged: FittingWidget. onSelectCategory()	-	Category selector combo box
5	Variable: FittingWidget.cbModel CurrentIndexChanged: FittingWidget.onSelectModel()	Sasmodels.generate. load_kernel_module()	Model selector combo box
6	Variable: FittingWidget. cbStructureFactor CurrentIndexChanged: FittingWidget. onSelectStructureFactor()	Sasmodels.product.m ake_product_info()	Structure factor combo box
7	Variable: FittingWidget._model_model dataChanged: FittingWidget.onMainParamsChange() -> ViewDelegate.ModelViewDelegate.__init__()	Sasmodels.modelinfo. make_model_info()	Model parameter list
8	Variable: FittingWidget.cmdPlot Clicked: FittingWidget.onPlot -> plotRequestedSignal	-	Plot button
9	Variable: FittingWidget.cmdFit Clicked: FittingWidget.onFit -> FitThread -> (batch)FittingCompleted	Sascalc.fit.BumpsFitti ng.BumpsFit().fit()	Fit button
10	Variable: FittingWidget.cmdHelp Clicked: FittingWidget.onHelp()	-	Help button



Entry Points: Fitting Options

src/sas/qtgui/Perspectives/Fitting

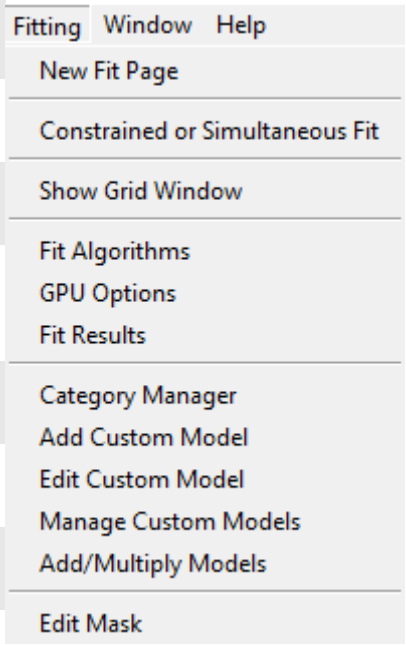
No.	qtgui	Description
1	OptionsWidget.__init__()	Options Widget – generated on SasView open
2	Variable: OptionsWidget.txtMinRange dataChanged: OptionsWidget.onModelChange()	Minimum Q for fit – updates when any options widget model item changes
3	Variable: OptionsWidget.txtMaxRange dataChanged: OptionsWidget.onModelChange()	Maximum Q for fit – updates when any options widget model item changes
4	Variable: OptionsWidget.cmdReset Clicked: OptionsWidget.onRangeReset()	Resets the Q range to the original range
5	Variable: OptionsWidget.cmdMaskEdit Clicked: OptionsWidget.onMaskEdit()	Opens a mask editor window to allow masking of data points – 2D only
6	Variable: OptionsWidget.txtNpts dataChanged: OptionsWidget.onModelChange()	Number of points selected – updates when any options widget model item changes
7	Variable: OptionsWidget.txtNptsFit dataChanged: OptionsWidget.onModelChange()	Number of points used for the fit – updates when any options widget model item changes
8	Variable: OptionsWidget.weightingGroup ButtonClicked: OptionsWidget.onWeightingChoice()	Intensity error weighting – dI becomes available if present in data, others when data is sent to fitting
9	Variable: FittingWidget.smearing_widget smearingChangedSignal: FittingWidget.onSmearingOptionsUpdate()	Resolution Tab Only option: Instrumental Smearing
10	Variable: FittingWidget.poly_params -> ViewDelegate.PolyViewDelegate.__init__()	Polydispersity Tab Disabled until Polydispersity box checked on Model tab
11	Variable: FittingWidget.lstMagnetic -> ViewDelegate.MagnetismViewDelegate.__init__()	Magnetism Tab – only for 2D data sets Disabled until Magnetism box checked on Model tab



Entry Points: Fitting Menu

src/sas/qtgui/
NB: This menu is only available when the Fitting perspective is active

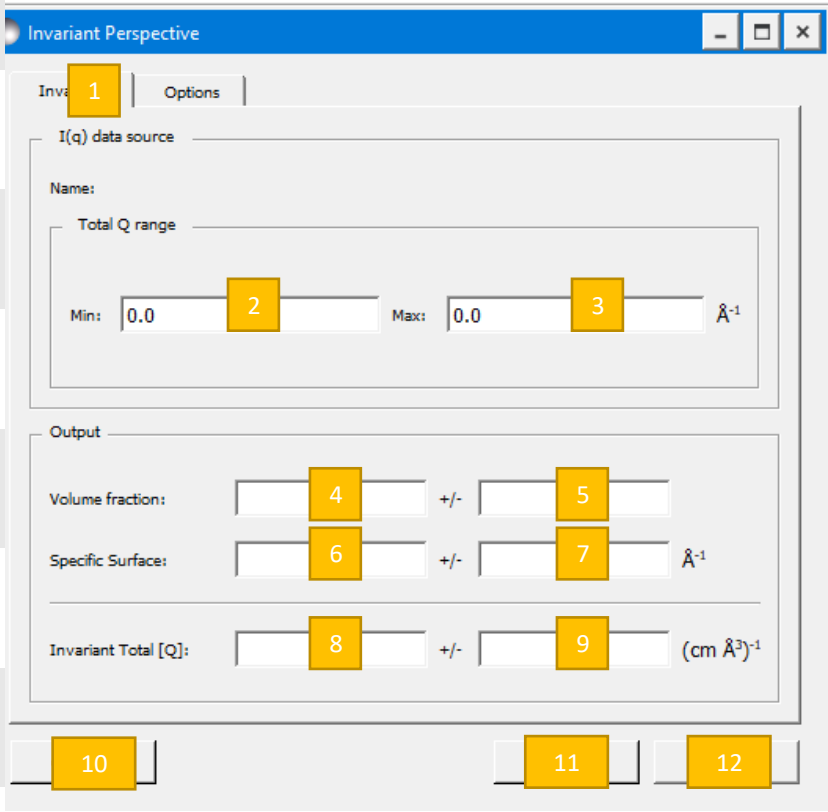
No.	qtgui	Description
New Fit Page	GuiManager.actionNew_Fit_Page() -> FittingPerspective.addFit()	Add a fit page to the fitting perspective
Constrained or Simultaneous Fit	GuiManager. actionConstrained_Fit() -> FittingPerspective.addConstraintTab()	Create a Constrained/Simultaneous fit page
Show Grid Window	GuiManager.actionCombine_Batch_Fit()	Open the Batch Fit results window
Fit Algorithms	GuiManager.actionFit_Options() -> FittingPerspective.fit_options_widget.show()	Open the Fit optimizer selection panel
GPU Options	GuiManager.actionGPU_Options() -> FittingPerspective.gpu_options_widget.show()	Open the GPU/CPU options panel
Fit Results	GuiManager.actionFit_Results() -> GuiManager.showFitResults() -> GuiManager.results_frame.setVisible(True)	Displays the fit results panel (DREAM optimizer only)
Category Manager	GuiManager.actionCategory_Manager() -> GuiManager.categoryManagerWidget.show()	Open the Model Category Manager CategoryManager()
Add Custom Model	GuiManager.actionAdd_Custom_Model() -> GuiManager.model_editor.show()	Open the model creation window Model_editor = TabbedModelEditor()
Edit Custom Model	GuiManager.actionEdit_Custom_Model() -> GuiManager.model_editor.show()	Open the custom model editor window Model_editor = TabbedModelEditor(edit_only=True)
Manage Custom Models	GuiManager.actionManage_Custom_Models() -> GuiManager.model_manager.show()	Open a list of custom models Model_manager = PluginManager
Add/Multiply Models	GuiManager.actionAddMult_Models() -> GuiManager.add_mult_editor.show()	Open the sum/multiply model window add_mult_editor = AddMultEditor()
Edit Mask	GuiManager.actionEditMask() -> extMaskEditorSignal.emit()	Open the 2D mask editor window



Entry Points: Invariant Perspective

src/sas/qtgui/Perspectives/Invariant
src/sas/sascalc/invariant/invariant.py

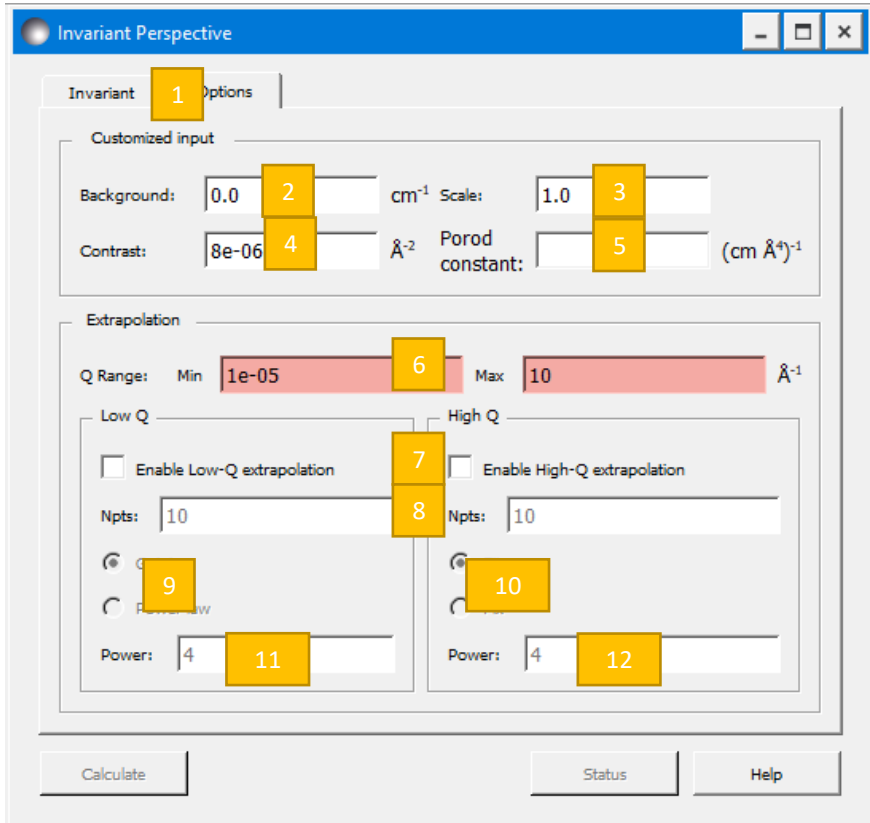
No.	qtgui	sascalc	Description
1	InvariantPerspective.__init__()	Variable: Invariant._calculator -> invariant.InvariantCalculator()	Invariant Perspective Tabs– generated on SasView open
2 & 3	Variable: txtTotalQMin Variable: txtTotalQMax	-	Minimum and Maximum Q From loaded data
4 & 5	Variable: txtVolFract Variable: txtVolFractErr	get_volume_fraction()	Calculated Volume Fraction and Error
6 & 7	Variable: txtSpecSurf Variable: txtSpecSurfErr	get_surface_with_error()	Calculated Specific Surface and Error
8 & 9	Variable: txtInvariantTot Variable: txtInvariantTotErr	get_qstar_with_error()	Calculated Total Invariant and Error
10	Variable: cmdCalculate calculateInvariant() -> calculateThread()	set_extrapolation(), get_qstar_low(), get_qstar_high()	Calculate Button Calculates the high and/or low Q extrapolation
11	Variable: cmdStatus onStatus() -> InvariantDetails.py	-	Invariant Details Button Opens window with bar chart with % for low, high, and total invariants
12	Variable: cmdHelp onHelp()	-	Help Button



Entry Points: Invariant Options

src/sas/qtgui/Perspectives/Invariant
src/sas/sascalc/invariant/invariant.py

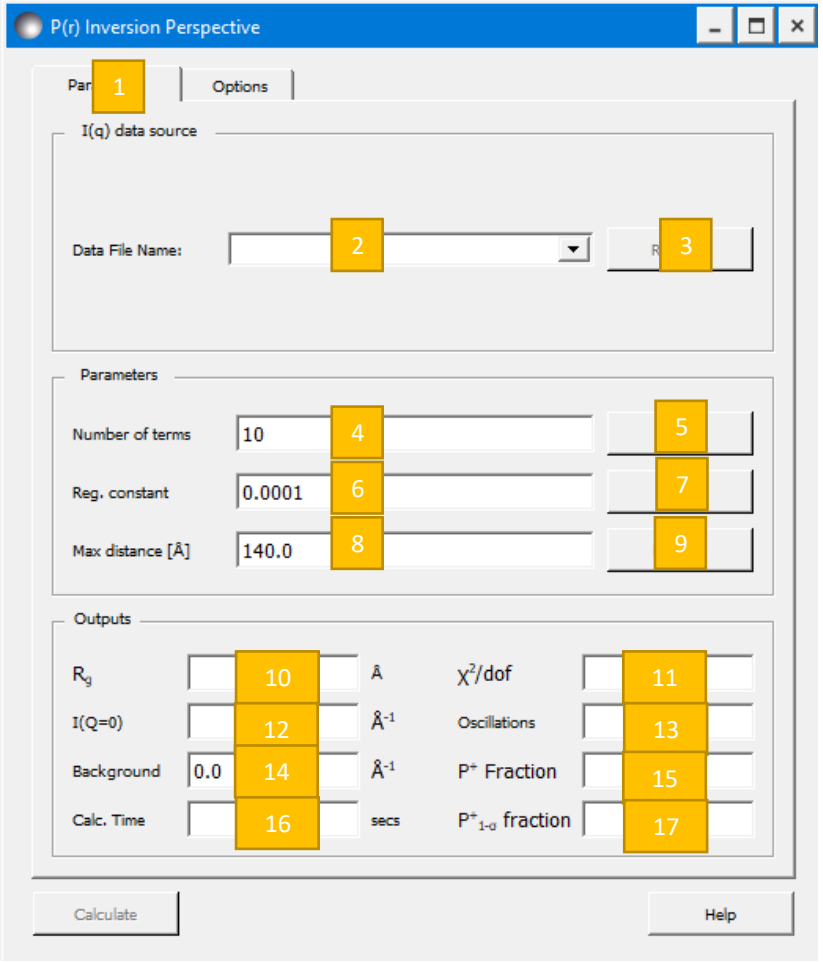
No.	qtgui	sascalc	Description
1	InvariantPerspective.__init__()	-	Invariant Perspective Tabs
2	Variable: txtBackgd textChanged: updateFromGui()	-	Background entered by user
3	Variable: txtScale textChanged: updateFromGui()	-	Scale factor entered by user
4	Variable: txtContrast textChanged: updateFromGui()	-	Contrast factor entered by user
5	Variable: txtScale textChanged: updateFromGui()	-	Porod constant entered by user
6	Variable: txtExtrapolQMin, txtExtrapolQMax textChanged: checkQMinRange(), checkQMaxRange()	-	Extrapolated Q range
7	Variable: chkLowQ, chkHighQ stateChanged: stateChanged(), checkQExtrapolatedData()	-	Enable Low and High Q ranges
8	Variable: txtNptsLowQm txtNptsHighQ textChanged: updateFromGui(), checkLength(), checkQRange()	-	Number of points used for low and high Q
9	Variable: rbGuinier, rbPowerLawLowQ Toggled: lowGuinierAndPowerToggle()	-	Low Q Guinier/Power Law Radio Buttons
10	Variable: rbFitHighQ Toggled: hiFitAndFixToggle()	-	High Q Fit/Fix Radio Buttons
11	Variable: txtPowerLowQ Textchanged: updateFromGui()	-	Low Q Guinier or Power Value
12	Variable: txtPowerHighQ Textchanged: updateFromGui()	-	Help Button



Entry Points: Inversion Perspective

src/sas/qtgui/Perspectives/Inversion
src/sas/sascalc/pr/

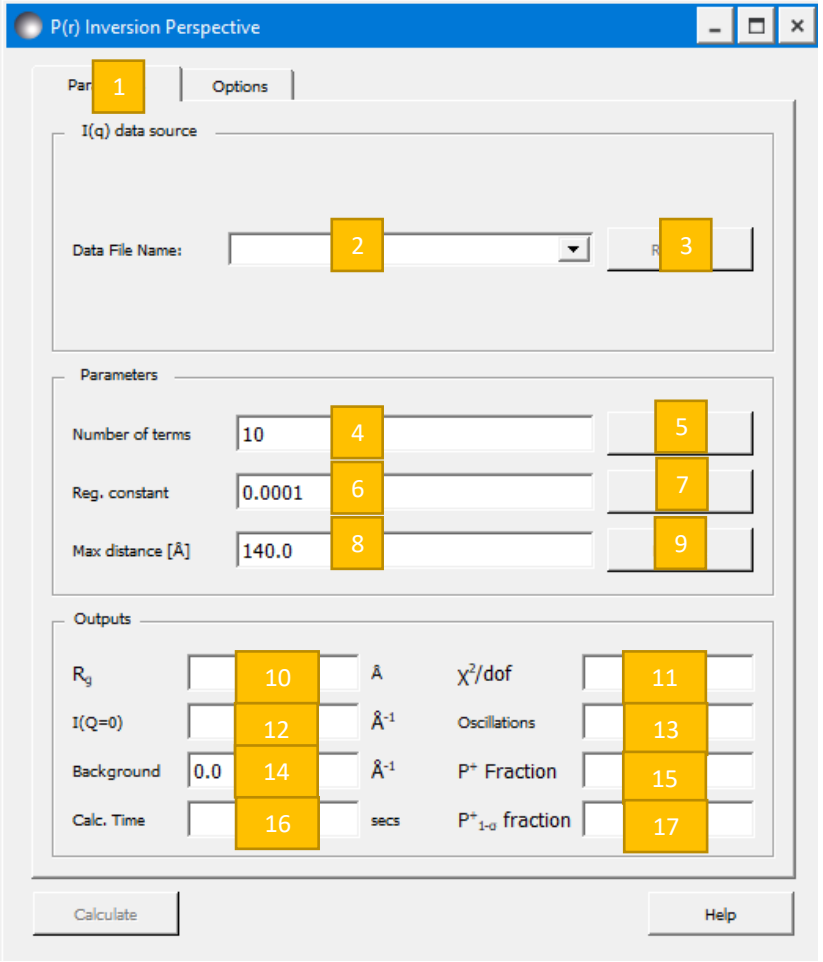
No.	qtgui	sascalc	Description
1	InversionPerspective.__init__()	Inverter.Invertor()	P(r) Inversion perspective – generated on SasView open, but hidden
2	dataList	-	Data file combo box (for future batch capabilities)
3	Variable: removeButton Clicked: removeData	-	Delete the data set selected in the combo box
4	noOfTermsInput	Invertor.nfunc	Number of terms used in the inversion calculation
5	noOfTermsSuggestionButton Clicked: acceptNoTerms	estimate_numterms()	Value is generated when data is loaded, and the estimate method is run
6	regularizationConstantInput	set_alpha()	Alpha/scaling factor
7	regConstantSuggestionButton Clicked: acceptAlpha	estimate_alpha()	Value is generated when data is loaded, and the estimate method is run
8	maxDistanceInput	set_dmax()	Maximum real space distance
9	explorerButton Clicked: openExplorerWindow()	-	Opens the data explorer window to show correlations between parameters



Entry Points: Inversion Perspective

src/sas/qtgui/Perspectives/Inversion
src/sas/sascalc/pr/

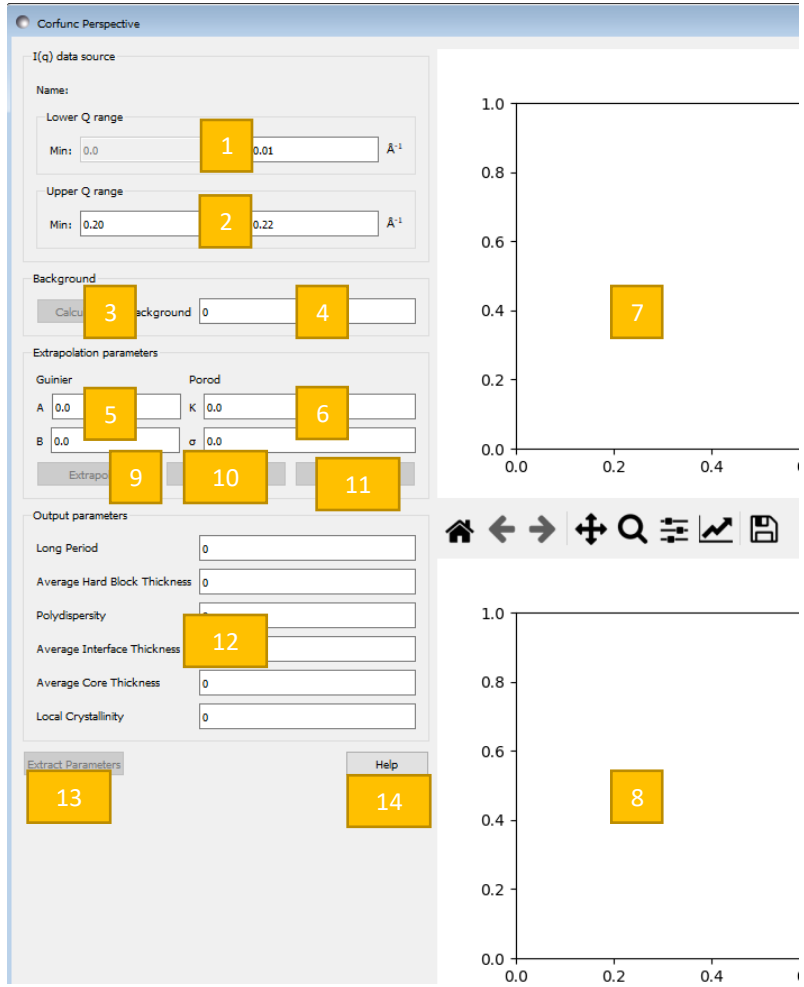
No.	qtgui	sascalc	Description
10	rgValue	Rg()	Calculated radius of gyration
11	chiDofValue	Chi2	Calculated chi^2
12	iQ0Value	Iq0()	Calculated I(Q=0)
13	oscillationValue	oscillations()	Oscillation parameters for the P(r)
14	backgroundValue	Background	Calculated background value for
15	posFractionValue	get_positive()	Calculated fraction of the P(r) > 0
16	sigmaPosFractionValue	get_pos_err()	Calculated fraction of P(r) the is 1-sigma greater than 0
17	computationTimeValue	elapsed	Computation time in seconds



Entry Points: Corfunc Perspective

src/sas/qtgui/Perspectives/Corfunc
src/sas/sascalc/corfunc/

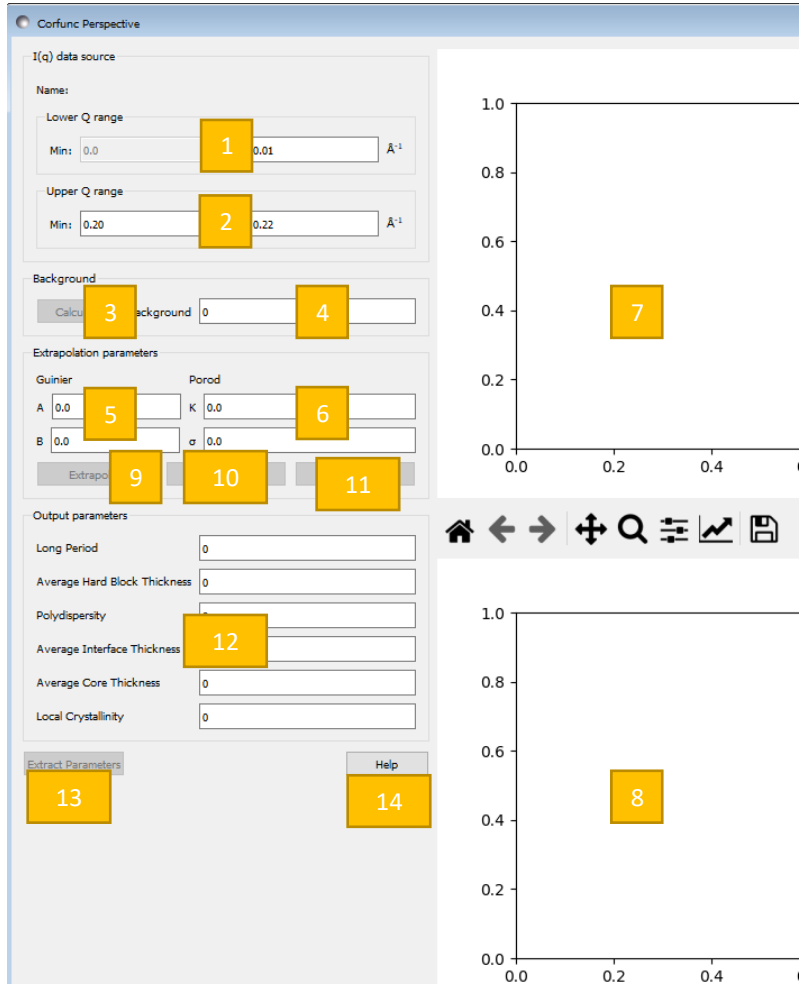
No.	qtgui	sascalc	Description
	Basis: CorfuncWindow.__init__()	Basis: CorfuncCalculator()	Corfunc Perspective Generated and hidden on loading
1	txtLowerQMin, txtLowerQMax _update_calculator()	0.0, min(_data.x) Linked to slider on plot	Lower Q Range
2	txtUpperQMin, txtUpperQMax _update_calculator()	Linked to slider on plot	Upper Q Range
3	cmdCalculateBg Clicked: calculate_background()	Compute_background()	Background calculation button
4	txtBackground _update_calculator()	background	Calculated background
5	txtGuinierA, txtGuinierB	Calculated in extrapolate()	Guinier Parameters A and B $I(Q) = Ae^{Bq^2}$
6	txtPorodK, txtPorodSigma	Calculated in extrapolate()	Porod Parameters $I(Q) = Kq^{-4}e^{-q^2sigma^2} + Bg$
7	_canvas	-	SAS Data Plot
8	_realplot	-	Real Space Plot



Entry Points: Corfunc Perspective

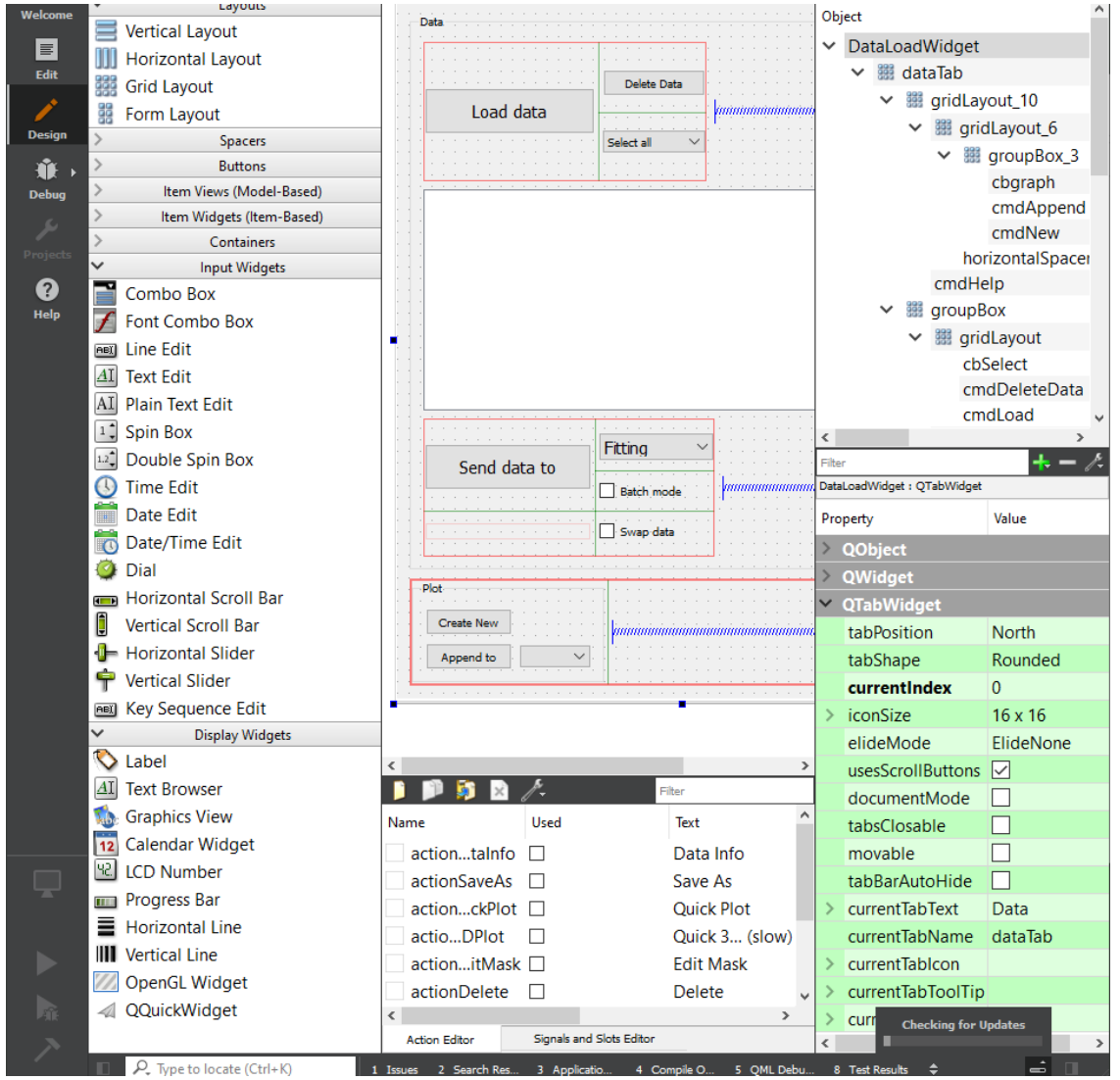
src/sas/qtgui/Perspectives/Corfunc
src/sas/sascalc/corfunc/

No.	qtgui	sascalc	Description
9	cmdExtrapolate Clicked: extrapolate()	compute_extrapolation()	Extrapolate Button
10	cmdTransform Clicked: transform()	Threaded: compute_transform()	Transform Button – Fourier transform high and low Q data
11	cmdSave Clicked: on_save()	-	Save data as a multi-column text file
12	txtAvgCoreThick, txtAvgIntThick, txtAvgHardBlock, txtPolydisp, txtLongPeriod, txtLocalCrystal	extract_parameters()	Correlation parameters Calculated when 'Extract' button is clicked
13	cmdExtract Clicked: extract()	extract_parameters()	Extract correlation parameters button
14	cmdHelp Clicked: showHelp()	-	Help button



Adding GUI elements

- Qt uses an XML format (.UI) to define the layout of each element
  - Qt Creator (free) can be used to edit and/or create new GUIs, or XML can be edited natively
- Housed in qtgui[/subpackage]/UI folders
- src/sas/qtgui/convertUI.py generates a python file from the XML, giving each element and interaction a unique python name
  - .py files in UI directories are included in .gitignore (**/UI/*.py)



Unit testing[1]

Sasview and sasmodels must be in python path!

- Tip: Create a symbolic link from git repo into python library

Three test locations

- Sascalc: Sasview/test
  - Run_one.py -> python run_one relative/path/to/test/file
    - Only a single file allowed per run
  - utest_sasview.py -> python utest_sasview.py
    - Runs all python files starting with 'utest'
- GUI: Sasview/src/sas/qtgui
  - GUITests.py -> python GUITests.py [suitename1 suitename2 …]
    - No suite options provided? Run them all!
    - Available suites are listed in GUITests.ALL_SUITES
  - *Failing tests (Github #1732)
- Model tests: sasmodels/sasmodels
  - Model_test.py -> python –m sasmodels.model_test [opencl|cuda|dll|all] model1 [model2 …]
  - 'all' instead of model1 […] will test all models

[1] https://github.com/SasView/sasview/wiki/Unit-testing-suites

Scriptable portions

- Sasmodels
  - Import sasmodels
  - Sasmodels.generate <modelname> # Generates DLL model
  - Sasmodels.compare [options] # Compares models to one another – see documentation for usage
- Data Loading (sasdata package in the works)
  - from sas.sascalc.dataloader.loader import Loader
  - Loader = Loader() # Not a callable class
  - File = Loader.load(<filePath>) # Returns [dataloader.data_info.Data(1|2)D()]
- Fitting
  - Possible, but more complicated
  - Generate model, load data, generate bumps fit, generate fit engine, tie all together and run fit()
- More to come…
  - Requires cleaner separation of calculations and GUI
  - Also requires API for other tools