| All variables defined in `*cry_ini.py*` | Found in `*Mtd.pref*` ? | Used Where? |
|---|---|---|
| RawDir | True | `def focus_all(…)`<br>`sampleSumLists = cry_utils.get_sample_list(EXPR_FILE.basefile, samplelistTexte, EXPR_FILE.RawDir)` |
| VanDir | True | - |
| VEmptyDir | True | - |
| SEmptyDir | True | - |
| CorrVanDir | True | `def load_sac_eff(EXPR_FILE, NoSAC=False, Eff=True):`<br>`newCalFile = EXPR_FILE.CorrVanDir + '/' + EXPR_FILE.GrpFile` |
| GrpDir | True | - |
| VanFile | False | `def load_sac_eff(EXPR_FILE, NoSAC=False, Eff=True):`<br>`(dum, uampstotal) = cry_sample.get_data_sum(EXPR_FILE.VanFile, "Vanadium", EXPR_FILE)`<br><br>`def create_vana(EXPR_FILE, NoAbs=False):`<br>`(dum, uampstotal) = cry_sample.get_data_sum(EXPR_FILE.VanFile, "Vanadium", EXPR_FILE)` |
| VEmptyFile | False | `def create_vana(EXPR_FILE, NoAbs=False):`<br>`(dum, uampstotal) = cry_sample.get_data_sum(EXPR_FILE.VEmptyFile, "Empty", EXPR_FILE)` |
| SEmptyFile | False | `if EXPR_FILE.SEmptyFile[0] != "none":`<br>`  Minus(LHSWorkspace="sample", RHSWorkspace="Sempty", OutputWorkspace="sample")`<br>`  mtd.remove("Sempty")` |
| AutoVan | False | - |
| CorrVanFile | True | `def focus_all(…)`<br><br>`vanfil = EXPR_FILE.CorrVanFile + "-" + str(spec) + ".nxs"`<br><br>`def create_vana(EXPR_FILE, NoAbs=False):`<br>`SaveNexusProcessed(Filename=EXPR_FILE.CorrVanFile + "_unstripped.nxs", InputWorkspace="Vanadium")`<br>`SaveFocusedXYE(Filename=EXPR_FILE.CorrVanFile + "_unstripped.dat", InputWorkspace="Vanadium", SplitFiles=True)`<br><br>`def save_vana(EXPR_FILE):`<br>`vanfil = EXPR_FILE.CorrVanFile + "-" + str(spec) + ".nxs"`<br><br>`def remove_bins(EXPR_FILE):`<br>`SaveFocusedXYE(Filename=EXPR_FILE.CorrVanFile + "-" + str(spec) + "_.dat", InputWorkspace="Vanadium-" + str(i),` |

| | | |
|---|---|---|
| | | SplitFiles=False)<br><br>def van_strip(EXPR_FILE):<br>SaveFocusedXYE(Filename=EXPR_FILE.CorrVanFile + "-" + str(spec) + "_.dat", InputWorkspace="Vanadium-" + str(i),<br>SplitFiles=False)<br><br>def van_spline(EXPR_FILE):<br>SaveFocusedXYE(Filename=EXPR_FILE.CorrVanFile + "-" + str(spec) + "_.dat", InputWorkspace="Vanadium-" + str(i),<br>SplitFiles=False)<br><br>def van_spline_only(EXPR_FILE):<br>SaveFocusedXYE(Filename=EXPR_FILE.CorrVanFile + "-" + str(spec) + "_.dat", InputWorkspace="Vanadium-" + str(i), SplitFiles=False) |
| SacEffFile | | - |
| ExistV | False | def focus_all(…)<br><br>if EXPR_FILE.ExistV == "load":<br><br>def create_vana(EXPR_FILE, NoAbs=False):<br>if EXPR_FILE.ExistV == 'no' and EXPR_FILE.VGrpfocus == 'van':<br>EXPR_FILE.write_prefline("ExistingV", "yes")<br>EXPR_FILE.ExistV = "yes" |
| VGrpfocus | False | def focus_one(…)<br>if EXPR_FILE.VGrpfocus == "sam" and isfirst:<br>cry_vana.create_vana(EXPR_FILE, NoAbs)<br><br>def create_vana(EXPR_FILE, NoAbs=False):<br>if EXPR_FILE.ExistV == 'no' and EXPR_FILE.VGrpfocus == 'van':<br>EXPR_FILE.write_prefline("ExistingV", "yes")<br>EXPR_FILE.ExistV = "yes" |
| Path2DatGrpFile | False | def focus_one(…)<br>EXPR_FILE.Path2DatGrpFile = newCalFile<br>...<br>DiffractionFocussing(InputWorkspace="sample", OutputWorkspace="sample", GroupingFileName=EXPR_FILE.Path2DatGrpFile,<br>PreserveEvents=False)<br><br>def create_vana(EXPR_FILE, NoAbs=False):<br>if EXPR_FILE.VGrpfocus == "sam":<br>GrpFile = EXPR_FILE.Path2DatGrpFile |
| VHeight | True | cry_utils.correct_abs(InputWkspc="Vanadium_corr", outputWkspc="Transmission", \<br>TheCylinderSampleHeight=EXPR_FILE.VHeight, \<br>TheCylinderSampleRadius=EXPR_FILE.VRadius, \<br>TheAttenuationXSection=EXPR_FILE.VAttenuationXSection, \<br>TheScatteringXSection=EXPR_FILE.VScatteringXSection, \<br>TheSampleNumberDensity=EXPR_FILE.VanaNumberDensity, \<br>TheNumberOfSlices=EXPR_FILE.VNumberOfSlices, \ |
| VRadius | True | |
| VAttenuationXSection | True | |
| VScatteringXSection | True | |
| VanaNumberDensity | True | |
| VNumberOfSlices | True | |
| VNumberOfAnnuli | True | |
| VNumberOfWavelengt | True | |

| | | |
|---|---|---|
| hPoints<br>VExpMethod | <br>True | TheNumberOfAnnuli=EXPR_FILE.VNumberOfAnnuli, \<br>TheNumberOfWavelengthPoints=EXPR_FILE.VNumberOfWavelengthPoints, \TheExpMethod=EXPR_FILE.VExpMethod) |
| VanSmooth | True | def remove_bins(EXPR_FILE):<br>OutputWorkspace="Vanadium-" + str(i),<br>NPoints=int(EXPR_FILE.VanSmooth)) |
| VanSplineCoef | True -> Differs | def van_spline(EXPR_FILE):<br>SplineBackground(InputWorkspace="Vanadium-" + str(i),<br>OutputWorkspace="Vanadium-" + str(i), WorkspaceIndex=0,<br>NCoeff=int(EXPR_FILE.VanSplineCoef))<br><br>def van_spline_only(EXPR_FILE):<br>SplineBackground(InputWorkspace="Vanadium-" + str(i),<br>OutputWorkspace="Vanadium-" + str(i), WorkspaceIndex=0,<br>NCoeff=int(EXPR_FILE.VanSplineCoef)) |
| VanPeakRemove | True -> Differs | def strip_the_vana(EXPR_FILE, LoadUnstrip=""):<br>if EXPR_FILE.VanPeakRemove == "interpol":<br>print " => Van Bragg-peak stripping"<br>print "Smmoth Vana data with " + EXPR_FILE.VanSmooth + " points"<br>remove_bins(EXPR_FILE)<br>elif EXPR_FILE.VanPeakRemove == "strip":<br>print " => Van Bragg-peak stripping"<br>van_strip(EXPR_FILE)<br>elif EXPR_FILE.VanPeakRemove == "spline":<br>van_spline(EXPR_FILE)<br>elif EXPR_FILE.VanPeakRemove == "splineonly":<br>van_spline_only(EXPR_FILE)<br>else:<br>return<br>save_vana(EXPR_FILE) |
| VanPeakFile | True | - |
| VanPeakList | False | def remove_bins(EXPR_FILE):<br>for peak in EXPR_FILE.VanPeakList[spec]:<br>RemoveBins(InputWorkspace="Vanadium-" + str(i),<br>OutputWorkspace="Vanadium-" + str(i), XMin=peak[0],<br>XMax=peak[1], RangeUnit="AsInput",<br>Interpolation="Linear", WorkspaceIndex=0)<br><br>def van_spline(EXPR_FILE):<br>for peak in EXPR_FILE.VanPeakList[spec]:<br>MaskBins(InputWorkspace="Vanadium-" + str(i),<br>OutputWorkspace="Vanadium-" + str(i), XMin=peak[0],<br>XMax=peak[1]) |
| VanPeakWdt<br>VanPeakTol | False<br>False | def van_strip(EXPR_FILE):<br>if EXPR_FILE.VanPeakWdt[spec] != 0:<br>EXPR_FILE.VanPeakWdt[spec]) + " Tol=" +<br>str(EXPR_FILE.VanPeakTol[spec])<br><br>StripPeaks(InputWorkspace="Vanadium-" + str(i),<br>OutputWorkspace="Vanadium-" + str(i),<br>FWHM=EXPR_FILE.VanPeakWdt[spec],<br>Tolerance=EXPR_FILE.VanPeakTol[spec], WorkspaceIndex=0)<br>SaveFocusedXYE(…) |
| CorrectSampleAbs | True | def focus_one(…) |

| | | |
|---|---|---|
| SampleAbsCorrected | False | `if EXPR_FILE.CorrectSampleAbs == "yes":`<br>`  if EXPR_FILE.SampleAbsCorrected == False:`<br>`  cry_utils.correct_abs(InputWkspc="sample", outputWkspc="SampleTrans", \`<br>`TheCylinderSampleHeight=EXPR_FILE.SampleHeight, \`<br><br>`TheCylinderSampleRadius=EXPR_FILE.SampleRadius, \`<br>`TheAttenuationXSection=EXPR_FILE.SampleAttenuationXSection, \`<br>`TheScatteringXSection=EXPR_FILE.SampleScatteringXSection, \`<br>`TheSampleNumberDensity=EXPR_FILE.SampleNumberDensity, \`<br>`TheNumberOfSlices=EXPR_FILE.SampleNumberOfSlices, \`<br>`TheNumberOfAnnuli=EXPR_FILE.SampleNumberOfAnnuli, \`<br>`TheNumberOfWavelengthPoints=EXPR_FILE.SampleNumberOfWavelengthPoints, \`<br>`TheExpMethod=EXPR_FILE.SampleExpMethod)`<br><br>`  EXPR_FILE.SampleAbsCorrected = True`<br>`else:`<br>`  ConvertUnits(…)`<br>`  Divide(…)`<br>`  ConvertUnits(…)`<br>`  DiffractionFocussing(…)` |
| SampleHeight | True | |
| SampleRadius | True | |
| SampleAttenuationXSection | True | |
| SampleScatteringXSection | True | |
| SampleNumberDensity | True | |
| SampleNumberOfSlices | True | |
| SampleNumberOfAnnuli | True | |
| SampleNumberOfWavelengthPoints | True | |
| SampleExpMethod | True | |
| LowerLambda | True | `def load_sac_eff(EXPR_FILE, NoSAC=False, Eff=True):`<br>`Integration(InputWorkspace="Eff", OutputWorkspace="Eff", \`<br>`  RangeLower=EXPR_FILE.LowerLambda, RangeUpper=EXPR_FILE.UpperLambda)` |
| UpperLambda | True -> Differs | |
| OutSuf | False | `def focus_one(…)`<br>`if EXPR_FILE.OutSuf == "":`<br>`  OutputFile = join(EXPR_FILE.user, outname)`<br>`else:`<br>`  OutputFile = join(EXPR_FILE.user, outname + "_" + EXPR_FILE.OutSuf)`<br>`# Gss`<br>`rearrang4gss(OutputFile, EXPR_FILE)`<br>`# Nexus`<br>`rearrange_4nex(OutputFile, EXPR_FILE)` |
| XYEDspc | False | - |
| XYEtof | False | - |
| Drange | False | `def focus_one(…)`<br>`cry_load.bin_bank("ResultD", EXPR_FILE.bankList, EXPR_FILE.Drange)`<br><br>`def bin_bank(InputArea, bankList, Drange):`<br>`  for i in bankList:`<br>`  Rebin(InputWorkspace=InputArea + "-" + str(i), OutputWorkspace=InputArea + "-" + str(i), Params=Drange[i - 1])`<br><br>`def sets_drange(wkspc, EXPR_FILE):`<br>`Drange = xbegin + ",-" + EXPR_FILE.Bining[i] + "," + xend`<br>`EXPR_FILE.Drange.append(Drange)` |

**Parameters:**

| **All variables read from** `mtd.pref` | **Equals `=` in** `mtd.pref` | **Used Where** |
|---|---|---|
| OffFile / Offsets | hrpd_new_072_01_corr .cal | - |
| GrpFile / Grouping | hrpd_new_072_01_corr .cal | `def load_sac_eff(…):`<br>`  # Loads SAC/Efficiency correction in wkspc "Corr" or sets it to "1"`<br>`  newCalFile = EXPR_FILE.CorrVanDir + '/' + EXPR_FILE.GrpFile`<br><br>`def focus_one(…):`<br>`  newCalFile = join(EXPR_FILE.user, EXPR_FILE.GrpFile)`<br><br>`def create_vana(EXPR_FILE, NoAbs=False):`<br>`if EXPR_FILE.VGrpfocus == "sam":`<br>`  GrpFile = EXPR_FILE.Path2DatGrpFile`<br>`else:`<br>`  GrpFile = EXPR_FILE.Path2VanGrpFile`<br>`. . .`<br>`DiffractionFocussing(InputWorkspace="Vanadium_corr", OutputWorkspace="Vanadium_foc", GroupingFileName=GrpFile, PreserveEvents=False)` |
| VrunnoList / Vanadium | 39191 | - |
| VErunnoList / V-Empty | 39187 | - |
| SErunnoList / S-Empty | 0 | - |
| ExistV / ExistingV | Yes | `def focus_all(…)`<br><br>`if EXPR_FILE.ExistV == "load":`<br>`  for i in EXPR_FILE.bankList:`<br>`  spec = i - 1`<br>`  vanfil = EXPR_FILE.CorrVanFile + "-" + str(spec) + ".nxs"`<br>`  LoadNexusProcessed(Filename=vanfil, OutputWorkspace="Vanadium-" + str(i))`<br>`  # CORRECT`<br>`elif EXPR_FILE.ExistV == "no" and EXPR_FILE.VGrpfocus == "van":`<br>`  print "was here?"`<br>`  cry_vana.create_vana(EXPR_FILE, NoAbs=NoVabs)` |
| *XYEtof / XYE-TOF* | *Yes* | - |
| *XYEdspc / XYE-D* | *Yes* | - |
| GSS | yes | `def rearrang4gss(OutputFile, EXPR_FILE):`<br>`if EXPR_FILE.GSS == "no":`<br>`  return`<br>`if len(EXPR_FILE.bankList[1:]) > 1:`<br>`  SaveGSS(InputWorkspace="ResultTOFgrp”)`<br>`else:`<br>`  SaveGSS(InputWorkspace="ResultTOF-1")` |

| | | |
|---|---|---|
| Nex | Yes | `def rearrange_4nex(OutputFile, EXPR_FILE):`<br>`  if EXPR_FILE.Nex == "no":`<br>`  return`<br>`  SaveNexusProcessed(Filename=OutputFile + ".nxs", InputWorkspace="ResultTOFgrp")` |
| NBank | 3 | `def sets_drange(wkspc, EXPR_FILE):`<br>`  datamatrix = mtd[wkspc]`<br>`  for i in range(0, EXPR_FILE.Nbank):`<br><br>`def __init__(self, instr):`<br>`  self.nbank = NUM_BANK_DIC[self.sname]`<br><br>`def load(…)`<br>`blist = range(1, self.Instr.nbank + 1)` |
| CropRange / CropRange | 0.05 0.95 | `def sets_drange(wkspc, EXPR_FILE):`<br>`  datamatrix = mtd[wkspc]`<br>`  for i in range(0, EXPR_FILE.Nbank):`<br>`  x_data = datamatrix.readX(i)`<br>`  last = len(x_data) - 1`<br>`  CropRange = EXPR_FILE.CropRange[i].rstrip().split()`<br>`  xbegin = str(x_data[0] * (1 + float(CropRange[0])))`<br>`  xend = str(x_data[last] * float(CropRange[1]))`<br>`  datbin = math.exp(math.log(x_data[last] / x_data[0]) / last) - 1`<br>`  if datbin > float(EXPR_FILE.Bining[i]):`<br>`  print 'WARNING: Rebining in *pref file ' + EXPR_FILE.Bining[`<br>`  i] + ' is lower than diffraction focusing rebining step'`<br>`  print 'WARNING: Rebining Kept to be ' + str(datbin) + ' for bank ' + str(i + 1)`<br>`  EXPR_FILE.Bining[i] = str(datbin)`<br>`  Drange = xbegin + ",-" + EXPR_FILE.Bining[i] + "," + xend`<br>`  EXPR_FILE.Drange.append(Drange)`<br>`  EXPR_FILE.dataRangeSet = True`<br>`  return sets_drange` |
| Bining / Bining | 0.0001 | |
| bankList / BankList | 1-3 | `def focus_one(…)`<br>`cry_load.bin_bank("ResultD", EXPR_FILE.bankList, EXPR_FILE.Drange)`<br>`. . .`<br>`cry_load.bin_bank("ResultD", EXPR_FILE.bankList, EXPR_FILE.Drange)`<br>`for i in EXPR_FILE.bankList:`<br>`  ConvertUnits()`<br>`  ReplaceSpecialValues()`<br>`  ReplaceSpecialValues()`<br><br>`def divide_samp_vana(EXPR_FILE, Norm):`<br>`for i in EXPR_FILE.bankList:`<br>`  RebinToWorkspace(`<br>`  Divide()`<br>`else:`<br>`  for i in EXPR_FILE.bankList:`<br>`  RenameWorkspace()` |

| | | |
|---|---|---|
| | | `def strip_the_vana(EXPR_FILE, LoadUnstrip=""):`<br>`  cry_load.split_bank("Vanadium", bankList=EXPR_FILE.bankList, Del=True)`<br><br>`def rearrange_4xye(…):`<br>`  if (units == "D" and EXPR_FILE.saveXYEd) or (units == "TOF" and EXPR_FILE.saveXYEtof):`<br>`  for i in EXPR_FILE.bankList:`<br>`  inwkpsc = "Result" + units + "-" + str(i)`<br>`  SaveFocusedXYE()`<br><br>`def split_bank(InputArea, bankList, Del=True):`<br>`  for i in bankList:`<br>`  CropWorkspace()`<br>`  if Del:`<br>`  mtd.remove(InputArea)`<br><br>`def bin_bank(InputArea, bankList, Drange):`<br>`  for i in bankList:`<br>`  Rebin()`<br><br>Many other places too… mostly as a loop in which which n algorithm is applied |