

VQS Breaking Changes

-  Breaking changes with the implementation of VivoScript (VQScript) by Release Version. Please also note public Deprecation list where we signal intent to remove elements of VQScript: [VivoQuant: Deprecated List](#)

6.0.1 Release

Prior to this Release	Changed with this Release	Notes
<pre>Array< string > saveImage(string filename, int objectType, int colBarType=0, int dcmRepType=0, int magnification=1, string cornerLabelTemplateName ="")</pre>	<pre>Array< string > saveImage(string filename, int objectType, int colBarType=0, int width = 0, int height = 0, int magnification=1, string cornerLabelTemplateName="")</pre>	<code>dcmRepType</code> removed, this would trigger the secondary capture mode which is now done through a complete separate API: <code>saveImageSecondaryCapture</code>. Also, width and height were added for more fine grained output sizing Refer to VQS documentation for details on the width, height and magnification Also, although it was undocumented before, <code>objectType = 4</code> has been switch with <code>objectType = 5</code> in behavior. <code>objectType = 4</code> now is

		all views separately. <code>objectType = 5</code> now is all views in one image
<pre>Array< string > saveMovie(string filename, int objectType, int colBarType=0, int dcmRepType=0, string cornerLabelTemplateName ="")</pre>	<pre>Array< string > saveMovie(string filename, int objectType, int colBarType=0, string cornerLabelTemplateName="")</pre>	<code>dcmRepType</code> removed, this would trigger the secondary capture mode which is now done through a complete separate API Also, although it was undocumented before, <code>objectType = 4</code> has been switch with <code>objectType = 5</code> in behavior. <code>objectType = 4</code> now is all views separately. <code>objectType = 5</code> now is all views in one image (only supported when used the Time series operator mode variant)

6.0 Release

	Prior to this Release	Changed with this Release	Notes
1	<code>VQ.currentView()</code> defaulted to slice view	<code>VQ.currentView()</code> has the ability to be <code>null</code> so the user/VQS writer must first make sure they have selected the view they want.	VQS writers should be aware of setting their chosen view and also checking if undefined.
2	<code>VQ.controler()</code> <code>VQ.mipControler()</code>	<code>VQ.controller()</code> <code>VQ.mipController()</code>	Name change; correct spelling of controller.

3	<code>VQ.mainWin().setViewMode</code> acted as deactivate/activate refresh	<code>VQ.mainWin().setViewMode()</code> now simply returns if the same view mode is already active.	Breaking change example is that <code>QuantTable</code> settings need to be set before the ROI operator activates as that's when they are read in. <code>setViewMode</code> will now not be used as a general refresh for these values to be ingested.
4	<code>VQ.setDebuggerEnabled()</code>		Removed as Qt6 does not have a replacement.
5	<code>VQ.controller().setIntensity()</code>	<code>VQ.controller().setIntensity()</code>	Renamed for clarity
6	<code>VQ.controller().getIntensity()</code>	<code>VQ.controller().getIntensity()</code>	Renamed for clarity
7	<code>VQ.buildID()</code>		This now returns the VivoQuant semantic version more reliably.
8	<code>VQ.storeData()</code> integer based indexes no longer index into 'flattened' timepoint indexes	Same interface - user should use the provided <code>index(imgIdx, timepointIdx)</code> if they want timepoints	The index argument in this function now indexes the top level nodes in the <code>DataManager</code> . Previously, this function indexed into timepoint volumes. For example, if a user had a 10 timepoint dynamic image loaded, and then a single timepoint image, index <code>1</code> did not give the user the image associated with the second entry in the <code>DataManager</code> , instead the user would get the second timepoint of the first image.

			Users who wish to get timepoint indexes should use <code>index(imgIdx, timepointIdx)</code> to properly get a two-level index.
9	<pre>VQ.storeData(AbstractDicomRepository *rep, Index idx, int format=-1) VQ.storeDataByTimepoint(AbstractDicomRepository *rep, int idx, int format=0, bool compress=false, int timepointOption=0, int timepointPerFileOption=0) VQ.storeAsDicom(string fn, Index idx, int format=-1, bool compress=false, bool quiet=true) VQ.storeAsDICOMByTimepoint(string fn, int idx, int format=0, bool compress=false, int timepointOption=0, int timepointPerFileOption=0)</pre>	In all of these APIs you must now set the format explicitly. Please refer to the latest VQS manual to see the DICOMImageFormat options.	Previously the default format was hooked up to the last saved format. This could have had unintended saved output formats. Being explicit removes all confusion in expectations.
10	<pre>VQ.dataManager().getMin()</pre>	<code>VQ.dataManager().updateAndGetMin()</code>	Clarified name of method to indicate the function is not purely a getter. A calculation

			will be run when this is called.
1 1	<code>VQ.dataManager().getMax()</code>	<code>VQ.dataManager().updateAndGetMax()</code>	Clarified name of method to indicate the function is not purely a getter. A calculation will be run when this is called.
1 2	<code>VQ.dataManager().getMinMax()</code>	<code>VQ.dataManager().updateAndGetMinMax()</code>	Clarified name of method to indicate the function is not purely a getter. A calculation will be run when this is called.
1 3	<code>VQ.dataManager().setDescription()</code>	Removed in favor of the user looping over their tags and calling <code>VQ.dataManager().setDescription()</code>	Reduce duplication of functionality.
1 4	<code>VQ.dataManager().fillIDICOMFromDesc()</code>	Removed	Removed unused function.
1 5	<code>VQ.workList().openTaskForm(const QString& taskID)</code> <code>VQ.workList().openTaskForm(int index)</code> <code>VQ.waitForECRF()</code> <code>VQ.setECRFCallback()</code> <code>VQ.clearCallback()</code> <code>VQ.eCRFCallback()</code> <code>VQ.eCRF()</code> <code>VQ.eCRFDone()</code> <code>VQ.eCRFError()</code> <code>VQ.workList().openTaskForm(const QString& taskID)</code>	Removed	The eCRF interface is no longer scriptable.

	VQ.workList().openTaskForm(int index) VQ.workList().getECRF() VQ.workList().eCRFDone() VQ.workList().isEditable() VQ.workList().getPrevTaskID()		
1 6	VQ.ipacs()	Removed	This function created a key-value store for login credentials that were not usable by an iPACS object instance. Instead, users should use Javascript objects if they need a key-value store for a similar use.
1 7	VQ.vtkController().forceVolumeMapperMode	VQ.vtkController().forceRenderingMode()	Duplicated other functionality.
1 8	VQ.vtkController().setPatientsName()	VQ.vtkController().setCorner0rEdgeLabel	VivoQuant enhanced its MIP labeling support in 5.X and doesn't need the old way to write the patient name to lower left corner.
1 9	VQ.mainWin().switchPatientName()	VQ.mainWin().switchCornerInfo()	Duplicated other functionality.
2 0	VQ.mainWin().switchSliceNumber()	VQ.mainWin().switchCornerInfo()	Duplicated other functionality.
2 1	VQ.dataManager().signalDataChange(int i)	Removed	
2 2	VQ.dataManager().signalDataUpdated()	VQ.dataManager().onDataNodeUpdated()	Renamed

2 3	<code>VQ.dataManager().signal0 ptEnabled(int i, bool b)</code>	Removed	
2 4	<code>VQ.dataManager().signalD ataLoaded()</code>	<code>VQ.dataManager().onDa taNodeLoaded()</code>	Renamed
2 5	<code>VQ.dataManager().signalD ataChanged()</code>	<code>VQ.dataManager().onDa taNodeChanged()</code>	Renamed
2 6	<code>VQ.dataManager().signalD ataUpdated()</code>	<code>VQ.dataManager().onDa taNodeUpdated()</code>	Renamed
2 7	<code>VQ.dataManager().signalD ataUnloaded()</code>	<code>VQ.dataManager().onDa taNodeUnloaded()</code>	Renamed
2 8	<code>VQ.dataManager().signalU pdateView()</code>	<code>Q.dataManager().onDat aChanged()</code>	Renamed
2 9	When ReorientationOperator is open <code>VQ.currentOp().deform0ve rlayToggled(int buttonIdx)</code> <code>VQ.currentOp().deform0ve rlayToggled2D(int buttonIdx)</code>	Removed	Removed unused functions.
3 0	<code>VQ.vtkViewer().resetCame raVeiwLocation(camNormal normal = camfront)</code>	<code>VQ.vtkViewer().resetC ameraViewOrientation(camNormal normal = camfront)</code>	Renamed
3 1	<code>VQ.PreProcessingTool().g etInjectedDose(int idx = 0)</code>	<code>VQ.PreProcessingTool() .getImgInjectedDoseV alue(int idx = 0) VQ.PreProcessingTool() .getImgInjectedDoseU nit(int idx = 0)</code>	Clarified that the fetched value will come from the image data instead of from the PreProcessing tool, and separated value and unit getters.
3 2	<code>VQ.PreProcessingTool().s etInjectedDose(string</code>	<code>VQ.PreProcessingTool() .setInjectedDoseValu</code>	Separated value and unit setters.

	dose)	e(string dose) VQ.PreProcessingTool().setInjectedDoseUnit (string dose)	
3 3	VQ.PreProcessingTool().getWeight(int idx = 0)	VQ.PreProcessingTool().getImgWeightValue(int idx = 0) VQ.PreProcessingTool().getImgWeightUnit(int idx = 0)	Clarified that the fetched value will come from the image data instead of from the PreProcessing tool, and separated value and unit getters.
3 4	VQ.PreProcessingTool().setWeight(string weight)	VQ.PreProcessingTool().setWeightValue(string dose) VQ.PreProcessingTool().setWeightUnit(string dose)	Separated value and unit setters.
3 5	VQ.openNMCT()	VQ.BrainAtlasTool()	Improved consistency with other handling of VivoQuant tools.
3 6	VQ.setNMCTQuietModeOn()	VQ.BrainAtlasTool().setQuietMode(bool quiet)	
3 7	VQ.setNMCTOutputFormat(string format)	VQ.BrainAtlasTool().setOutputFormat(string format)	
3 8	VQ.runNMCT()	VQ.BrainAtlasTool().run()	
3 9	VQ.getDouble(...) VQ.getDoubleInteractive(...) VQ.getInt(...) VQ.getIntInteractive(...)	VQ.getValues(string title, Object map, bool modal = true)	Consolidated interface.

<pre>VQ.getItem(...)</pre> <pre>VQ.getItemInteractive(...)</pre> <pre>VQ.getText(...)</pre> <pre>VQ.getTextInteractive(...)</pre>		
---	--	--

5.4 Release

Prior to this Release	Changed with this Release	Notes
	No Changes	

5.3 Release

Prior to this Release	Changed with this Release	Notes
	No Changes	

5.2 Release

Prior to this Release	Changed with this Release	Notes
<pre>mw.switchAutoZoom(true)</pre> <pre>;</pre>	<pre>mw.setAndApplyAutoZoom(true)</pre> <pre>);</pre>	We've made it so that Auto-Zoom does not lock other zoom options. This saves you a click so you no longer have to turn auto-zoom off before zooming in or out. Associated with VQ-3909.