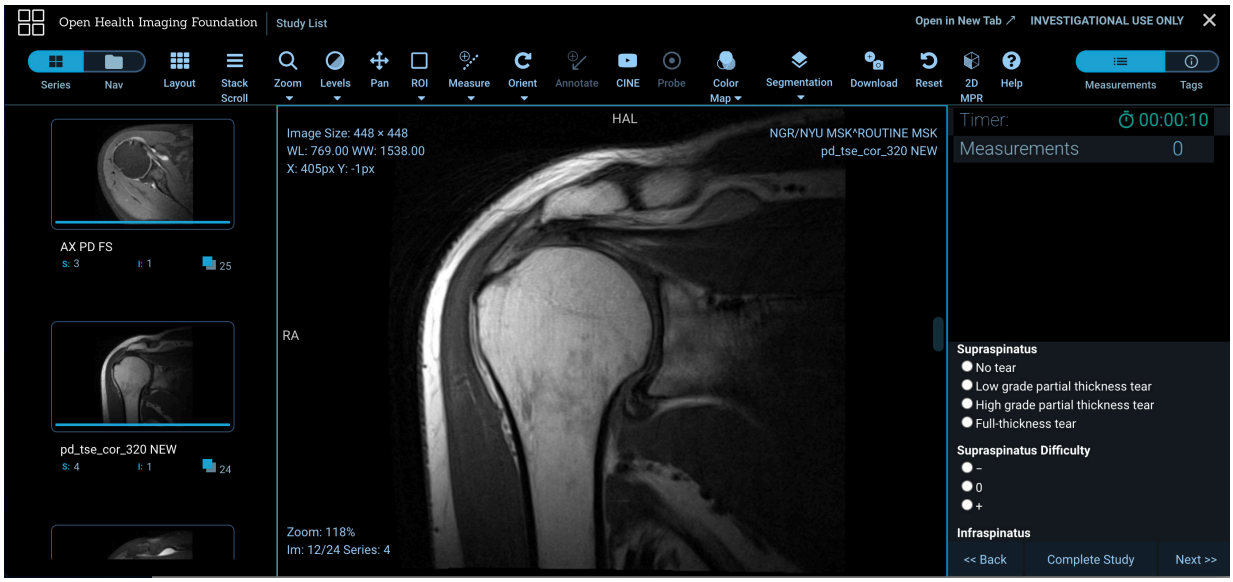
The OHIF config file

From creating your own labels to controlling the hanging protocol, the `ohif_config.json` file is a powerful tool for customizing your Flywheel workflows. Additionally, you can also buildyour own blind reader study questions to display alongside images. Using the integrated viewer and study forms allows you to automatically capture responses to form questions alongside the images themselves.

This guide describes the configuration options in the `ohif_config.json` file.

You can also download a pdf of the guide.



Viewer Settings

These settings configure the look and feel of the viewer includingFrom hiding toolbar options, to creating custom labels for ROIS, and configuring keyboard shortcuts.

- **general settings**: includes enabling timers, hiding measurements, and configuring how multiple contours interact.
- **toolbar**: hide tools that do not apply to your workflow
- **labels**: Add custom labels along with limits on how many times each label should be used
- **layouts**: Configure the hanging protocol for the viewer
- **hotkeys**: Configure keyboard shortcuts inside the viewer
- **mouseActions**: Configure what mouse actions for inside the viewer

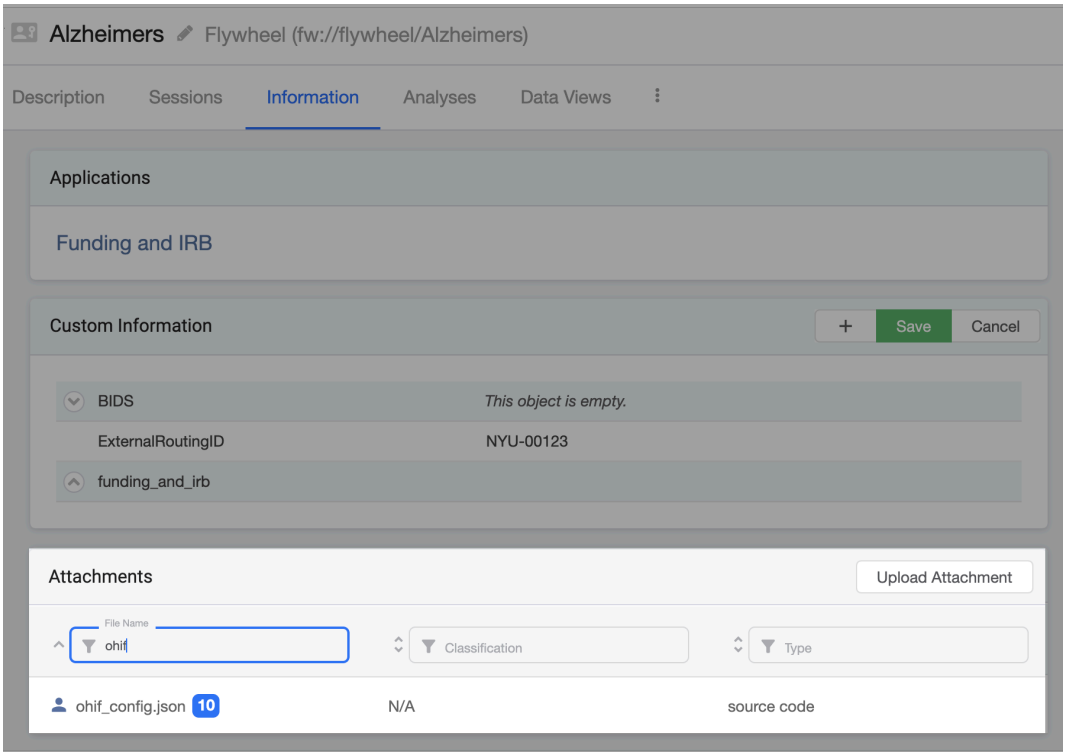
Build a study form

Create questions for readers to answer using the studyForm configurations. Learn more about studyForm options.

Applying the OHIF configurations

To apply these configurations to your project:

1. Run the file through a linter such as the Online JSON linter to make sure you have valid JSON.
2. Save the files as **ohif_config.json.**
3. Go to the Information tab for your project, and upload it as an attachment



General settings

These are the top level of settings in the `ohif_config.json` file

Property	Description	Value	Required
resetTool	Reset the active tool when loading a new study	boolean	no
toolbar	Configure available tools in the toolbar	`object` See below for more details	no

Property	Description	Value	Required
labels	Custom labels for measurements in the OHIF Viewer	Array of objects with the following properties. • **label** - The text displayed as a measurement option • **value** - The key to store measurement data under • **limit** - The number of allowed measurements using this label per study • **boundary**-Sets a boundary for the contour so that distinct features cannot overlap See the labels section for more details	no
studyForm	Options for creating a study form	`object` See the studyForm section for more details	no
hideMeasurements	Hide measurements in the right panel • If `studyForm` and its components are not included and `hideMeasurements: true`, no study summary/form will be displayed but measurements will be • If `studyForm` and its components are not included and `hideMeasurements: false` , no study summary/form or measurements will be displayed	boolean	no
layouts	Define the hanging protocol for grid and asymmetric layout	object See layouts section for more details	no
allowDraft	Enable or Disable the draft feature in study form	boolean	no
draftStudy		boolean	no
hideFreehandVertex	Vertex points are not displayed on the image	boolean	no

Property	Description	Value	Required
hideMeasurementOutputs	Measurement output are not displayed on the image	boolean	no
hideROIoutput	ROI area, mean, std dev are not displayed on the image	boolean	no
timerOn	Run timer until reader moves to the next session. Total “read time” is captured and saved along with form input as metadata on the session	boolean	no
timerVisible	Displays timer for the reader in the right panel	boolean	no
contourUnique	When set to true, contours/ROI’s cannot overlap on the same slice	boolean	no
automatedLabelling	Automatically applies a label to a measurement. This setting is enable by default. Set to false to disable. Automate labeling only applies in study form questions where both the `requireMeasurements` and the `measurementTools` are configured.	boolean	no

```
{
  "draftStudy": true,
  "resetTool": false,
  "hideFreehandVertex": true,
  "hideROIoutput": true,
  "hideMeasurements": false,
  "timerOn": true,
  "timerVisible": true,
  "contourUnique": true,
  "automatedLabelling": true,
  "labels": [],
  "studyForm": [],
  "hotkeys": [],
  "toolbar": []
}
```

toolbar

The `toolbar` section allows the tool set to be trimmed down by category. Below is the default toolbar:

Category	Description	Tools
zoom	Zoom in/out	• **Zoom** • **Magnify**
measurements	ROI and Measure menus	• **EllipticalRoi** • **RectangleRoi** • **WwwcRegion** - ROI Window • **FreehandRoi** • **Eraser** • **Clear** • **Length** • **Bidirectional** • **Angle** • **ArrowAnnotate** See below for more details*

*Tool	Image
EllipicalRoi	
RectangleR-oi	
Freehan-dRoi	
WwwcRe-gion (ROI win-dow)	
Eraser	
Clear	

*Tool	Image
Length Bidirection-al Angel	
ArrowAnno-tate	

Each category can have an `only` or `except` array that allows or restricts available tools in that category.

Below shows an example of all of the tools you can hide in the toolbar along with how to hide them using the config :


```
"toolbar": {
    "Zoom": {
      "except": [
    "Zoom",
    "Magnify"
      ]
    },
    "Measure": {
      "except": [
        "Length",
        "Bidirectional",
        "Angle"
      ]
    },
    "ROI": {
      "except": [
```

```
      "Ellipse",
      "Rectangle",
      "ROI Window",
      "Freehand",
      "Open Freehand",
      "Sculpt",
      "Eraser",
      "Clear"
    ]
  },
  "Segmentation": {
    "except": [
      "Segmentation"
    ]
  },
  "Reset": {
    "except": [
      "Reset"
    ]
  },
  "Annotate": {
    "except": [
      "Annotate"
    ]
  },
  "CINE": {
    "except": [
      "CINE"
    ]
  },
  "Protocols": {
    "except": [
      "Protocols"
    ]
  },
  "Download": {
```

```
      "except": [
        "Download"
      ]
    },
    "2D MPR": {
      "except": [
        "2D MPR"
      ]
    },
    "Help": {
      "except": [
        "Help"
      ]
    },
    "Link": {
      "except": [
        "Scroll",
        "Levels",
         "Zoom"
      ]
    },
    "Crosshairs": {
      "except": [
        "Crosshairs"
      ]
    }
  }
```

labels

Create custom labels and configure how they interact with other labels

Property	Description	Value
label	What appears to the reader as the label as well as how to reference the label	string

Property	Description	Value
value	The key to store measurement as metadata	string
limit	The number of allowed measurements using this label per study	number
boundary	Sets a boundary for the contour so that distinct features cannot overlap	• label: the label that should be used as a boundary • direction: which direction the boundary applies. Options for open contours are up, down, left, right Options for closed contours are: • **Inward**: label with boundary defintion (alpha) needs to be fully encompassed in the other label noted (beta) Example: If beta was the rib cage and alpha was the heart, the heart is contained inside the chest cavity • **Outward**: label with boundary defintion (alpha) can not be encompassed in the other label noted (beta) // aka anywhere else in the image Example: If beta was the liver and alpha was the kidney, they are two separate organs that can't be inside one another

Example labels with boundaries:

```
{
  "zoomLevel": {
    "minimum": 0.25,
    "maximum": 20
  },
  "contourUnique": true,
  "labels": [
    {
      "label": "Bruch's Membrane",
      "value": "BM",
      "limit": 1
    },
    {
      "label": "Supraspinatus - anteroposterior",
      "value": "SAP",
      "limit": 1,
```

```
          "boundary":  [{"label": "Bruch's Membrane",
"direction": "inward"}]
    },
    {
      "label": "Supraspinatus - mediolateral",
      "value": "SML",
      "limit": 2,
          "boundary":  [{"label": "SAP", "direction":
"outward"}]
    },
    {
      "label": "Retinal Pigment Epithelium",
      "value": "RPE",
      "limit": 1
    },
    {
      "label": "Photoreceptor Layers",
      "value": "PR",
      "limit": 1,
      "boundary": [
        {
          "label": "Retinal Pigment Epithelium",
          "direction": "inward"
        }
      ]
    },
    {
      "label": "External Limiting Membrane",
      "value": "ELM",
      "limit": 1,
      "boundary": [
        {
          "label": "PR",
          "direction": "up/down"
        }
      ]
```

```
    },
    {
      "label": "Ellipsoid Zone",
      "value": "EZ",
      "limit": 1,
          "boundary":  [{"label": "Retinal Pigment
Epithelium", "direction": "left/right"}]
    },
    {
      "label": "Outer Plexifrom Layer - Henle Fiber Layer",
      "value": "OPL-HFL",
      "limit": 1
    },
    {
      "label": "Inner Limiting Membrane",
      "value": "ILM",
      "limit": 1
    },
    {
      "label": "Intraretinal Fluid",
      "value": "IRF",
      "limit": 1
    },
    {
      "label": "Subretinal Fluid",
      "value": "SRF",
      "limit": 1
    },
    {
      "label": "Subretinal Hyperreflective Material",
      "value": "SHRM",
      "limit": 1
    },
    {
      "label": "Pigmented Epithelium Detachment -
Fibrovascular",
```

```
      "value": "PED-F",
      "limit": 1
    },
    {
      "label": "Pigmented Epithelium Detachment - Serous",
      "value": "PED-S",
      "limit": 1
    },
    {
      "label": "Pigmented Epithelium Detachment - Drusenoid",
      "value": "PED-D",
      "limit": 1
    }
  ],
  "hotkeys": [
    {
      "commandName": "incrementActiveViewport",
      "label": "Next Viewport",
      "keys": [
        "right"
      ]
    },
    {
      "commandName": "decrementActiveViewport",
      "label": "Previous Viewport",
      "keys": [
        "left"
      ]
    },
    {
      "commandName": "rotateViewportCW",
      "label": "Rotate Right",
      "keys": [
        "r"
      ]
    },
```

```
  {
    "commandName": "rotateViewportCCW",
    "label": "Rotate Left",
    "keys": [
      "l"
    ]
  },
  {
    "commandName": "invertViewport",
    "label": "Invert",
    "keys": [
      "i"
    ]
  },
  {
    "commandName": "flipViewportVertical",
    "label": "Flip Horizontally",
    "keys": [
      "h"
    ]
  },
  {
    "commandName": "flipViewportHorizontal",
    "label": "Flip Vertically",
    "keys": [
      "v"
    ]
  },
  {
    "commandName": "scaleUpViewport",
    "label": "Zoom In",
    "keys": [
      "+"
    ]
  },
  {
```

```
    "commandName": "scaleDownViewport",
    "label": "Zoom Out",
    "keys": [
      "-"
    ]
  },
  {
    "commandName": "fitViewportToWindow",
    "label": "Zoom to Fit",
    "keys": [
      "="
    ]
  },
  {
    "commandName": "resetViewport",
    "label": "Reset",
    "keys": [
      "space"
    ]
  },
  {
    "commandName": "nextImage",
    "label": "Next Image",
    "keys": [
      "down"
    ]
  },
  {
    "commandName": "previousImage",
    "label": "Previous Image",
    "keys": [
      "up"
    ]
  },
  {
    "commandName": "previousViewportDisplaySet",
```

```
    "label": "Previous Series",
    "keys": [
      "pagedown"
    ]
  },
  {
    "commandName": "nextViewportDisplaySet",
    "label": "Next Series",
    "keys": [
      "pageup"
    ]
  },
  {
    "commandName": "deleteMeasurement",
    "label": "Delete Measurement",
    "keys": [
      "backspace",
      "del"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "Zoom"
    },
    "label": "Zoom",
    "keys": [
      "z"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "Wwwc"
    },
    "label": "Wwwc",
```

```
      "keys": [
        "w",
        "esc"
      ]
    },
    {
      "commandName": "setActiveToolHotkey",
      "commandOptions": {
        "toolName": "Magnify"
      },
      "label": "Magnify",
      "keys": [
        "m"
      ]
    },
    {
      "commandName": "setActiveToolHotkey",
      "commandOptions": {
        "toolName": "Pan"
      },
      "label": "Pan",
      "keys": [
        "p"
      ]
    },
    {
      "commandName": "setActiveToolHotkey",
      "commandOptions": {
        "toolName": "StackScroll"
      },
      "label": "Stack Scroll",
      "keys": [
        "s"
      ]
    },
    {
```

```
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "Angle"
    },
    "label": "Angle",
    "keys": [
      "a"
    ]
  },
  {
    "commandName": "prevMeasureTool",
    "label": "Previous Measurement Tool",
    "keys": [
      "["
    ]
  },
  {
    "commandName": "nextMeasureTool",
    "label": "Next Measurement Tool",
    "keys": [
      "]"
    ]
  },
  {
    "commandName": "firstImage",
    "label": "First Image",
    "keys": [
      "home"
    ]
  },
  {
    "commandName": "lastImage",
    "label": "Last Image",
    "keys": [
      "end"
    ]
```

```
  },
  {
    "commandName": "nextTemporalSeries",
    "label": "Next Time",
    "keys": [
      "."
    ]
  },
  {
    "commandName": "previousTemporalSeries",
    "label": "Previous Time",
    "keys": [
      ","
    ]
  },
  {
    "commandName": "setFullDynamicWWWC",
    "label": "Full Dynamic W/L",
    "keys": [
      "0"
    ]
  },
  {
    "commandName": "windowLevelPreset1",
    "label": "W/L Preset 1",
    "keys": [
      "1"
    ]
  },
  {
    "commandName": "windowLevelPreset2",
    "label": "W/L Preset 2",
    "keys": [
      "2"
    ]
  },
```

```
  {
    "commandName": "windowLevelPreset3",
    "label": "W/L Preset 3",
    "keys": [
      "3"
    ]
  },
  {
    "commandName": "windowLevelPreset4",
    "label": "W/L Preset 4",
    "keys": [
      "4"
    ]
  },
  {
    "commandName": "windowLevelPreset5",
    "label": "W/L Preset 5",
    "keys": [
      "5"
    ]
  },
  {
    "commandName": "windowLevelPreset6",
    "label": "W/L Preset 6",
    "keys": [
      "6"
    ]
  },
  {
    "commandName": "windowLevelPreset7",
    "label": "W/L Preset 7",
    "keys": [
      "7"
    ]
  },
  {
```

```
      "commandName": "windowLevelPreset8",
      "label": "W/L Preset 8",
      "keys": [
        "8"
      ]
    },
    {
      "commandName": "windowLevelPreset9",
      "label": "W/L Preset 9",
      "keys": [
        "9"
      ]
    },
    {
      "commandName": "mpr2d",
      "label": "2D MPR",
      "keys": [
        "d"
      ]
    }
  ],
  "mouseActions": [
    {
      "toolName": "Rotate",
      "button": "middle"
    },
    {
      "toolName": "StackScroll",
      "button": "right"
    },
    {
      "toolName": "Crosshair",
      "button": "left"
    },
    {
      "toolName": "ZoomMouseWheel",
```

```
      "button": "wheel"
    }
  ]
}
```

layouts

Specify different hanging protocols and configure the following settings for what images appear as well as how they are displayed.

Property	Description	Value	Required
name	Any name for layout	string	yes
selector	Define tags and values for finding matching protocol for a study/series metadata	• **tag** - DICOM tags or property name • **match** - matching regular expression	yes
type	Define the type of layout	• **grid** - Grid type layout(Default) • **asymmetric** - Asymmetric type layout	no
position	Position of the viewport	• **x** - Horizontal position value in range 0 -1 • **y** - Vertical position value in range 0 -1	yes
size	Size of the viewport	• **width** - Horizontal size value in range 0 -1 • **height** - Vertical size value in range 0 -1	yes
type	Type of the viewport	• **2D** - 2D DICOM image viewport(Default) • **2D MPR** - Multi-Planar-Reconstructed viewport with 2D display • **3D** - 3D volume display(Not supported now)	no
orientation	Orientation for 2D MPR viewport	• **axial** - Axial MPR view • **sagittal** - Sagittal MPR view • **coronal** - Coronal MPR view • **Specify an array with 6 numbers** specifying the slice normal and view up direction co-ordinate values	yes
SeriesDescription	Find the matching series description	string You can also use REGEX pattern matching.	yes

Grid

For defining the Grid layout, we have to define the viewports as an array of array where the outer array defines the row count and inner array defines the column count. Below is a sample 1x2 (1 Row, 2 Columns) format

```
{
  "layouts": [
    {
      "name": "MR",
      "selector": {
        "tag": "00080060",
        "match": "^MR$"
      },
      "viewports": [
        [
          {
            "tag": "0008103E",
            "match": "^3D_PD.*"
          },
          {
            "tag": "0008103E",
            "match": "^3D_T2FS.*"
          }
        ]
      ]
    }]
}
```

asymmetrical

For defining the asymmetric layout, we have to define the viewports as an array by providing the position and size of each viewport. The asymmetric format type is an extended viewport definition where we can specify the type of viewport(2D and 2D MPR) and the orientation for 2D MPR type. The Grid layout can also be achieved by using this asymmetric format by providing proper position and size values. Below is some of the sample asymmetric layout format:

Grid layout with asymmetric layout type

```
{
  "layouts": [
    {
      "name": "MR",
```

```
      "selector": {
        "tag": "00080060",
        "match": "^.*"
      },
      "type": "asymmetric",
      "viewports": [
          {
                  "position": {"x": 0, "y": 0},
                  "size": {"width": 0.333, "height": 1},
                  "type": "2D MPR",
                  "orientation": "axial",
                  "SeriesDescription": "^.*"
          },
          {
                  "position": {"x": 0.333, "y": 0},
                  "size": {"width": 0.334, "height": 1},
                  "type": "2D MPR",
                  "orientation": "sagital",
                  "SeriesDescription": "^.*"
          },
          {
                  "position": {"x": 0.667, "y": 0},
                  "size": {"width": 0.333, "height": 1},
                  "type": "2D",
                  "SeriesDescription": "^.*"
          }
        ]
     }]
}
```

Asymmetric layout with different viewport dimension

```
{
  "layouts": [
    {
      "name": "Modality",
      "selector": {
```

```
        "tag": "00080060",
        "match": "^.*"
      },
      "type": "asymmetric",
      "viewports": [
          {
                  "position": {"x": 0, "y": 0},
                  "size": {"width": 0.75, "height": 1},
                  "type": "2D",
                  "SeriesDescription": "^.*"
          },
          {
                  "position": {"x": 0.75, "y": 0},
                  "size": {"width": 0.25, "height": 0.5},
                  "type": "2D MPR",
                  "orientation": "sagittal",
                  "SeriesDescription": "^.*"
          },
          {
                  "position": {"x": 0.75, "y": 0.5},
                  "size": {"width": 0.25, "height": 0.5},
                  "type": "2D MPR",
                  "orientation": "coronal",
                  "SeriesDescription": "^.*"
          }
       ]
    }
 ]
}
```

keyboard shortcuts and mouse actions

To better fit your readers' workflows, configure keyboard shortcuts (also known as hot keys) and mouse actions within the viewer.

hotKeys

property	Description	Value
commandName	Describe the action	string
label	label the shortcut	string
keys	Key for the action	The key used for the shortcut

```
{
"hotkeys": [
    {
      "commandName": "incrementActiveViewport",
      "label": "Next Viewport",
      "keys": [
        "right"
      ]
    },
    {
      "commandName": "decrementActiveViewport",
      "label": "Previous Viewport",
      "keys": [
        "left"
      ]
    },
    {
      "commandName": "rotateViewportCW",
      "label": "Rotate Right",
      "keys": [
        "r"
      ]
    },
    {
      "commandName": "rotateViewportCCW",
      "label": "Rotate Left",
      "keys": [
        "l"
      ]
    },
```

```
  {
    "commandName": "invertViewport",
    "label": "Invert",
    "keys": [
      "i"
    ]
  },
  {
    "commandName": "flipViewportHorizontal",
    "label": "Flip Horizontally",
    "keys": [
      "h"
    ]
  },
  {
    "commandName": "flipViewportVertical",
    "label": "Flip Vertically",
    "keys": [
      "v"
    ]
  },
  {
    "commandName": "scaleUpViewport",
    "label": "Zoom In",
    "keys": [
      "+"
    ]
  },
  {
    "commandName": "scaleDownViewport",
    "label": "Zoom Out",
    "keys": [
      "-"
    ]
  },
  {
```

```
    "commandName": "fitViewportToWindow",
    "label": "Zoom to Fit",
    "keys": [
      "="
    ]
  },
  {
    "commandName": "resetViewport",
    "label": "Reset",
    "keys": [
      "space"
    ]
  },
  {
    "commandName": "nextImage",
    "label": "Next Image",
    "keys": [
      "down"
    ]
  },
  {
    "commandName": "previousImage",
    "label": "Previous Image",
    "keys": [
      "up"
    ]
  },
  {
    "commandName": "previousViewportDisplaySet",
    "label": "Previous Series",
    "keys": [
      "pagedown"
    ]
  },
  {
    "commandName": "nextViewportDisplaySet",
```

```
    "label": "Next Series",
    "keys": [
      "pageup"
    ]
  },
  {
    "commandName": "deleteMeasurement",
    "label": "Delete Measurement",
    "keys": [
      "backspace",
      "del"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "Zoom"
    },
    "label": "Zoom",
    "keys": [
      "z"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "Wwwc"
    },
    "label": "Wwwc",
    "keys": [
      "w",
      "esc"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
```

```
    "commandOptions": {
      "toolName": "Magnify"
    },
    "label": "Magnify",
    "keys": [
      "m"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "Pan"
    },
    "label": "Pan",
    "keys": [
      "p"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "StackScroll"
    },
    "label": "Stack Scroll",
    "keys": [
      "s"
    ]
  },
  {
    "commandName": "setActiveToolHotkey",
    "commandOptions": {
      "toolName": "Angle"
    },
    "label": "Angle",
    "keys": [
      "a"
```

```
      ]
    },
    {
      "commandName": "prevMeasureTool",
      "label": "Previous Measurement Tool",
      "keys": [
        "["
      ]
    },
    {
      "commandName": "nextMeasureTool",
      "label": "Next Measurement Tool",
      "keys": [
        "]"
      ]
    },
    {
      "commandName": "firstImage",
      "label": "First Image",
      "keys": [
        "home"
      ]
    },
    {
      "commandName": "lastImage",
      "label": "Last Image",
      "keys": [
        "end"
      ]
    },
    {
      "commandName": "nextTemporalSeries",
      "label": "Next Time",
      "keys": [
        "."
      ]
```

```
  },
  {
    "commandName": "previousTemporalSeries",
    "label": "Previous Time",
    "keys": [
      ","
    ]
  },
  {
    "commandName": "setFullDynamicWWWC",
    "label": "Full Dynamic W/L",
    "keys": [
      "0"
    ]
  },
  {
    "commandName": "windowLevelPreset1",
    "label": "W/L Preset 1",
    "keys": [
      "1"
    ]
  },
  {
    "commandName": "windowLevelPreset2",
    "label": "W/L Preset 2",
    "keys": [
      "2"
    ]
  },
  {
    "commandName": "windowLevelPreset3",
    "label": "W/L Preset 3",
    "keys": [
      "3"
    ]
  },
```

```
  {
    "commandName": "windowLevelPreset4",
    "label": "W/L Preset 4",
    "keys": [
      "4"
    ]
  },
  {
    "commandName": "windowLevelPreset5",
    "label": "W/L Preset 5",
    "keys": [
      "5"
    ]
  },
  {
    "commandName": "windowLevelPreset6",
    "label": "W/L Preset 6",
    "keys": [
      "6"
    ]
  },
  {
    "commandName": "windowLevelPreset7",
    "label": "W/L Preset 7",
    "keys": [
      "7"
    ]
  },
  {
    "commandName": "windowLevelPreset8",
    "label": "W/L Preset 8",
    "keys": [
      "8"
    ]
  },
  {
```

```
      "commandName": "windowLevelPreset9",
      "label": "W/L Preset 9",
      "keys": [
        "9"
      ]
    },
    {
      "commandName": "mpr2d",
      "label": "2D MPR",
      "keys": [
        "d"
      ]
    }
  ]
}
```

mouseActions

Property	Description	Value
toolname	Tool from navigation bar to open	• Zoom • Pan • Rotate • StackScroll • Wwwc (ROI Window) • StackScrollMouseWheel • ZoomMouseWheel • Crosshair • RectangleRoi • EllipticalRoi • FreehandRoi
Button	part of mouse	• left • right • middle • wheel *These options depend on how you are viewing images. See the table below for more details

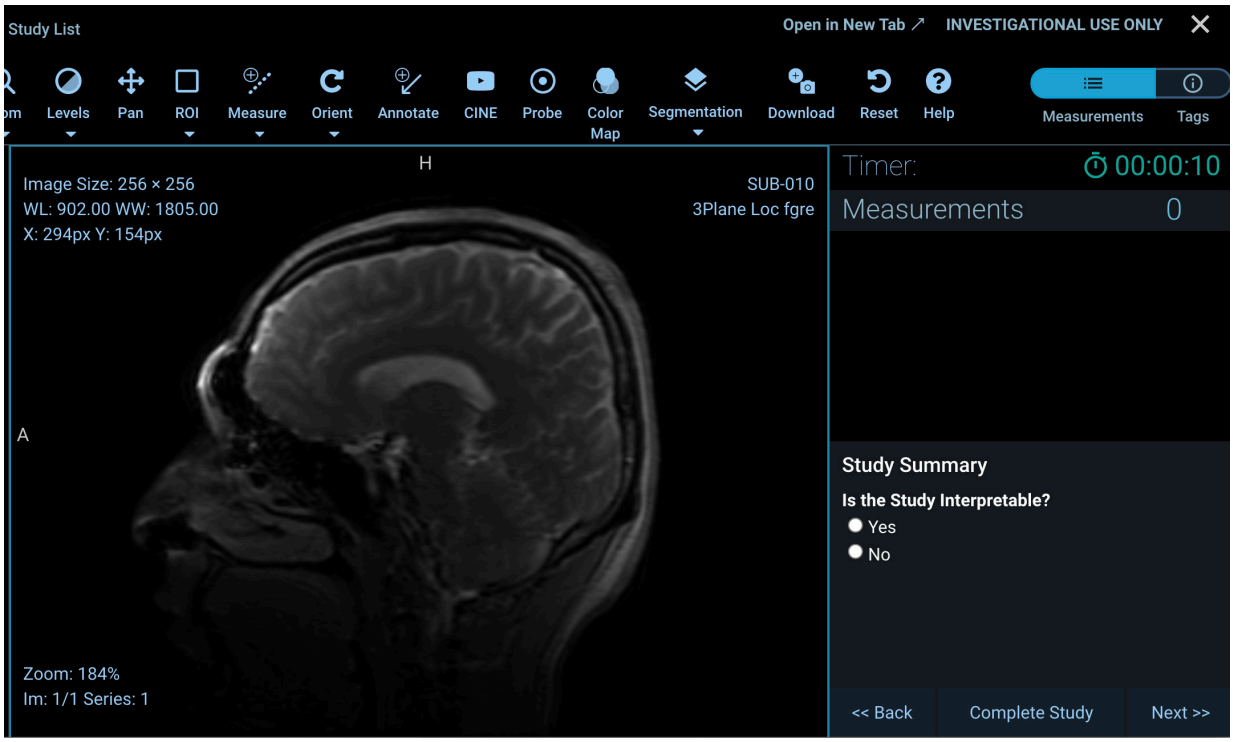
*Button	Tools available in 2d	Tools available in 2dmpr
left	Rotate, StackScroll, Zoom, Pan, Wwwc, and ROI tools	Rotate, StackScroll, Zoom, Pan, Wwwc, and Crosshairs

*Button	Tools available in 2d	Tools available in 2dmpr
right	Pan, Zoom, Rotate, StackScroll, and Wwwc	Pan, Zoom, Rotate
middle	Pan, Zoom, Rotate, StackScroll, and Wwwc	Pan, Zoom, Rotate
wheel	StackScrollMouseWheel, Zoom-MouseWheel	StackScrollMouseWheel, ZoomMouseWheel

```
{
  "mouseActions": [
    {
      "toolName": "Zoom",
      "button": "middle"
    },
    {
      "toolName": "Pan",
      "button": "right"
    },
    {
      "toolName": "RectangleRoi",
      "button": "left"
    },
    {
      "toolName": "StackScrollMouseWheel",
      "button": "wheel"
    }
  ]
}
```

studyForm

The studyForm section is where you can create blind reader studies. These questions can conditional based on the previous answer and you can configure certain tools and labels to appear based on answers. See below for an example set of questions.



Property	Description	Value	Required
key	Property name to save question to study Custom Information	string (no special characters other than - and _)	yes
values	Choices for `radio` and `selectboxes` type questions.	Array of objects with the following properties. • **label**: Text for question • **value**: Value to store if option is selected • **excludeMeasurements**: Don't render this option if the following measurements are selected • **requireMeasurements**: Render this option if the following measurements are selected	no
label	Question text	string	yes
required	Require an answer for this question	boolean	no

Property	Description	Value	Required
type	The type of question	options: • **radio** : radio buttons and labels for each option in values • **text**: DEPRECATED - See textarea • **textarea** : Text area with label • **textfield** : text input (single line) with label • **selectboxes** : Checkboxes and labels for each option in values • **content**: HTML Content. for example, adding a header to separate questions into different categories (use the html `html` key instead of the `label` key when using this)	yes
conditional	Condition under which to render this question. You can use the keys of other questions and values.	Object with the following properties. json - The JSON Logic to determine if this component is conditionally available.	no
validate	Conditions under which to validate this question.	Object with the following properties. required - If the field is required custom - A custom javascript based validation or a JSON object for using JSON Logic	no

Example

Below is an example of study form questions with conditional logic:

```
{
"studyForm": {
    "components": [
      {
        "label": "Is the Study Interpretable?",
        "key": "interpretable",
        "required": true,
        "type": "radio",
        "values": [
          {
            "value": "yes",
            "label": "Yes"
          },
          {
```

```
        "value": "no",
        "label": "No"
      }
    ]
  },
    {
    "html": "<h4>A Header</h4>",
    "key": "header1",
    "type": "content",
    "conditional": {
      "json": {
        "==": [
          {
            "var": "next1"
          },
          "yes"
        ]
      }
    }
  },

  {
    "label": "----------RETINAL LAYERS----------",
    "key": "retinal_layers",
    "conditional": {
      "json": {
        "==": [
          {
            "var": "interpretable"
          },
          "yes"
        ]
      }
    },
    "required": false,
    "type": "radio",
```

```
      "values": []
    },
    {
      "label": "Is BM present?",
      "key": "pres_BM",
      "conditional": {
        "json": {
          "==": [
            {
              "var": "interpretable"
            },
            "yes"
          ]
        }
      },
      "required": true,
      "type": "radio",
      "values": [
        {
          "value": "yes",
          "label": "Yes",
          "requireMeasurements": [
            "BM"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
        },
        {
          "value": "possible",
          "label": "Possible",
          "requireMeasurements": [
            "BM"
          ],
          "measurementTools": [
            "FreehandRoi"
```

```
          ]
        },
        {
          "value": "no",
          "label": "No"
        },
        {
          "value": "ungradable",
          "label": "Ungradable"
        }
      ]
    },
    {
      "label": "Contour complete - BM?",
      "key": "pres_BM_contour",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_BM"
                },
                "yes"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_BM"
                },
                "possible"
              ]
            }
          ]
        }
```

```
    },
    "required": true,
    "type": "radio",
    "values": [
      {
        "value": "yes",
        "label": "Yes"
      },
      {
        "value": "no",
        "label": "No"
      }
    ]
  },
  {
    "label": "Is SAP present?",
    "key": "pres_SAP",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_BM_contour"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
                "var": "pres_BM_contour"
              },
              "possible"
            ]
          },
```

```
        {
          "==": [
            {
              "var": "pres_BM"
            },
            "no"
          ]
        },
        {
          "==": [
            {
              "var": "pres_BM"
            },
            "ungradable"
          ]
        }
      ]
    }
  },
  "required": true,
  "contourVisibility": true,
  "type": "radio",
  "values": [
    {
      "value": "yes",
      "label": "Yes",
      "requireMeasurements": [
        "SAP"
      ],
      "measurementTools": [
        "Length"
      ]
    },
    {
      "value": "possible",
      "label": "Possible",
```

```
        "requireMeasurements": [
          "SAP"
        ],
        "measurementTools": [
          "Length"
        ]
      },
      {
        "value": "no",
        "label": "No"
      },
      {
        "value": "ungradable",
        "label": "Ungradable"
      }
    ]
  },
  {
    "label": "Contour complete - SAP?",
    "key": "pres_SAP_contour",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_SAP"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
                "var": "pres_SAP"
              },
```

```
              "possible"
            ]
          }
        ]
      }
    },
    "required": true,
    "type": "radio",
    "values": [
      {
        "value": "yes",
        "label": "Yes"
      },
      {
        "value": "no",
        "label": "No"
      }
    ]
  },
  {
    "label": "Is SML present?",
    "key": "pres_SML",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_SAP_contour"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
```

```
              "var": "pres_SAP_contour"
            },
            "possible"
          ]
        },
        {
          "==": [
            {
              "var": "pres_SAP"
            },
            "no"
          ]
        },
        {
          "==": [
            {
              "var": "pres_SAP"
            },
            "ungradable"
          ]
        }
      ]
    }
  },
  "required": true,
  "type": "radio",
  "values": [
    {
      "value": "yes",
      "label": "Yes",
      "requireMeasurements": [
        "SML"
      ],
      "measurementTools": [
        "Bidirectional",
        "Angle"
```

```
          ]
        },
        {
          "value": "possible",
          "label": "Possible",
          "requireMeasurements": [
            "SML"
          ],
          "measurementTools": [
            "Bidirectional",
            "Angle"
          ]
        },
        {
          "value": "no",
          "label": "No"
        },
        {
          "value": "ungradable",
          "label": "Ungradable"
        }
      ]
    },
    {
      "label": "Contour complete - SML?",
      "key": "pres_SML_contour",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_SML"
                },
                "yes"
              ]
```

```
            },
            {
              "==": [
                {
                  "var": "pres_SML"
                },
                "possible"
              ]
            }
          ]
        }
      },
      "required": true,
      "type": "radio",
      "values": [
        {
          "value": "yes",
          "label": "Yes"
        },
        {
          "value": "no",
          "label": "No"
        }
      ]
    },
    {
      "label": "Is RPE present?",
      "key": "pres_RPE",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_SML_contour"
                },
```

```
            "yes"
          ]
        },
        {
          "==": [
            {
              "var": "pres_SML"
            },
            "no"
          ]
        },
        {
          "==": [
            {
              "var": "pres_SML"
            },
            "ungradable"
          ]
        }
      ]
    }
  },
  "required": true,
  "type": "radio",
  "contourVisibility": true,
  "values": [
    {
      "value": "yes",
      "label": "Yes",
      "requireMeasurements": [
        "RPE"
      ],
      "measurementTools": [
        "FreehandRoi"
      ]
    },
```

```
        {
          "value": "possible",
          "label": "Possible",
          "requireMeasurements": [
            "RPE"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
        },
        {
          "value": "no",
          "label": "No"
        },
        {
          "value": "ungradable",
          "label": "Ungradable"
        }
      ]
    },
    {
      "label": "Contour complete - RPE?",
      "key": "pres_RPE_contour",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_RPE"
                },
                "yes"
              ]
            },
            {
              "==": [
```

```
                {
                  "var": "pres_RPE"
                },
                "possible"
              ]
            }
          ]
        }
      },
      "required": true,
      "type": "radio",
      "values": [
        {
          "value": "yes",
          "label": "Yes"
        },
        {
          "value": "no",
          "label": "No"
        }
      ]
    },
    {
      "label": "Is EZ present?",
      "key": "pres_EZ",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_RPE_contour"
                },
                "yes"
              ]
            },
```

```
        {
          "==": [
            {
              "var": "pres_RPE"
            },
            "no"
          ]
        },
        {
          "==": [
            {
              "var": "pres_RPE"
            },
            "ungradable"
          ]
        }
      ]
    }
  },
  "required": true,
  "type": "radio",
  "values": [
    {
      "value": "yes",
      "label": "Yes",
      "requireMeasurements": [
        "EZ"
      ],
      "measurementTools": [
        "FreehandRoi"
      ]
    },
    {
      "value": "possible",
      "label": "Possible",
      "requireMeasurements": [
```

```
            "EZ"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
        },
        {
          "value": "no",
          "label": "No"
        },
        {
          "value": "ungradable",
          "label": "Ungradable"
        }
      ]
    },
    {
      "label": "Contour complete - EZ?",
      "key": "pres_EZ_contour",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_EZ"
                },
                "yes"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_EZ"
                },
                "possible"
```

```
            ]
          }
        ]
      }
    },
    "required": true,
    "type": "radio",
    "values": [
      {
        "value": "yes",
        "label": "Yes"
      },
      {
        "value": "no",
        "label": "No"
      }
    ]
  },
  {
    "label": "Is ELM present?",
    "key": "pres_ELM",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_EZ_contour"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
                "var": "pres_EZ"
```

```
            },
            "no"
          ]
        },
        {
          "==": [
            {
              "var": "pres_EZ"
            },
            "ungradable"
          ]
        }
      ]
    }
  },
  "required": true,
  "type": "radio",
  "values": [
    {
      "value": "yes",
      "label": "Yes",
      "requireMeasurements": [
        "ELM"
      ],
      "measurementTools": [
        "FreehandRoi"
      ]
    },
    {
      "value": "possible",
      "label": "Possible",
      "requireMeasurements": [
        "ELM"
      ],
      "measurementTools": [
        "FreehandRoi"
```

```
          ]
        },
        {
          "value": "no",
          "label": "No"
        },
        {
          "value": "ungradable",
          "label": "Ungradable"
        }
      ]
    },
    {
      "label": "Contour complete - ELM?",
      "key": "pres_ELM_contour",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_ELM"
                },
                "yes"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_ELM"
                },
                "possible"
              ]
            }
          ]
        }
```

```
    },
    "required": true,
    "type": "radio",
    "values": [
      {
        "value": "yes",
        "label": "Yes"
      },
      {
        "value": "no",
        "label": "No"
      }
    ]
  },
  {
    "label": "Is OPL-HFL present?",
    "key": "pres_OPLHFL",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_ELM_contour"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
                "var": "pres_ELM"
              },
              "no"
            ]
          },
```

```
            {
              "==": [
                {
                  "var": "pres_ELM"
                },
                "ungradable"
              ]
            }
          ]
        }
      },
      "required": true,
      "type": "radio",
      "values": [
        {
          "value": "yes",
          "label": "Yes",
          "requireMeasurements": [
            "OPL-HFL"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
        },
        {
          "value": "possible",
          "label": "Possible",
          "requireMeasurements": [
            "OPL-HFL"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
        },
        {
          "value": "no",
```

```
        "label": "No"
      },
      {
        "value": "ungradable",
        "label": "Ungradable"
      }
    ]
  },
  {
    "label": "Contour complete - OPL HFL?",
    "key": "pres_OPLHFL_contour",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_OPLHFL"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
                "var": "pres_OPLHFL"
              },
              "possible"
            ]
          }
        ]
      }
    },
    "required": true,
    "type": "radio",
    "values": [
```

```
        {
          "value": "yes",
          "label": "Yes"
        },
        {
          "value": "no",
          "label": "No"
        }
      ]
    },
    {
      "label": "Is ILM present?",
      "key": "pres_ILM",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_OPLHFL"
                },
                "no"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_OPLHFL"
                },
                "ungradable"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_OPLHFL_contour"
```

```
            },
            "yes"
          ]
        }
      ]
    }
  },
  "required": true,
  "type": "radio",
  "values": [
    {
      "value": "yes",
      "label": "Yes",
      "requireMeasurements": [
        "ILM"
      ],
      "measurementTools": [
        "FreehandRoi"
      ]
    },
    {
      "value": "possible",
      "label": "Possible",
      "requireMeasurements": [
        "ILM"
      ],
      "measurementTools": [
        "FreehandRoi"
      ]
    },
    {
      "value": "no",
      "label": "No"
    },
    {
      "value": "ungradable",
```

```
        "label": "Ungradable"
      },
      {
        "label": "Contour complete - ILM?",
        "key": "pres_ILM_contour",
        "required": true,
        "type": "radio",
        "values": [
          {
            "value": "yes",
            "label": "Yes"
          },
          {
            "value": "no",
            "label": "No"
          }
        ]
      }
    ]
  },
  {
    "label": "--------------PATHOLOGY--------------",
    "key": "pathology",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_ILM_contour"
              },
              "yes"
            ]
          },
          {
            "==": [
```

```
              {
                "var": "pres_ILM"
              },
              "no"
            ]
          },
          {
            "==": [
              {
                "var": "pres_ILM"
              },
              "ungradable"
            ]
          }
        ]
      }
    },
    "required": false,
    "type": "radio",
    "values": []
  },
  {
    "label": "Is IRF present?",
    "key": "pres_path_IRF",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_ILM_contour"
              },
              "yes"
            ]
          },
          {
```

```
            "==": [
              {
                "var": "pres_ILM"
              },
              "no"
            ]
          },
          {
            "==": [
              {
                "var": "pres_ILM"
              },
              "ungradable"
            ]
          }
        ]
      }
    },
    "required": true,
    "type": "radio",
    "values": [
      {
        "value": "yes",
        "label": "Yes",
        "requireMeasurements": [
          "IRF"
        ],
        "measurementTools": [
          "FreehandRoi"
        ]
      },
      {
        "value": "possible",
        "label": "Possible",
        "requireMeasurements": [
          "IRF"
```

```
        ],
        "measurementTools": [
          "FreehandRoi"
        ]
      },
      {
        "value": "no",
        "label": "No"
      },
      {
        "value": "ungradable",
        "label": "Ungradable"
      }
    ]
  },
  {
    "label": "Contour complete - IRF?",
    "key": "pres_IRF_contour",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_path_IRF"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
                "var": "pres_path_IRF"
              },
              "possible"
            ]
```

```
            }
          ]
        }
      },
      "required": true,
      "type": "radio",
      "values": [
        {
          "value": "yes",
          "label": "Yes"
        },
        {
          "value": "no",
          "label": "No"
        }
      ]
    },
    {
      "label": "Is SRF present?",
      "key": "pres_path_SRF",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_IRF_contour"
                },
                "yes"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_path_IRF"
                },
```

```
            "no"
          ]
        },
        {
          "==": [
            {
              "var": "pres_path_IRF"
            },
            "ungradable"
          ]
        }
      ]
    }
  },
  "required": true,
  "type": "radio",
  "values": [
    {
      "value": "yes",
      "label": "Yes",
      "requireMeasurements": [
        "SRF"
      ],
      "measurementTools": [
        "FreehandRoi"
      ]
    },
    {
      "value": "possible",
      "label": "Possible",
      "requireMeasurements": [
        "SRF"
      ],
      "measurementTools": [
        "FreehandRoi"
      ]
```

```
        },
        {
          "value": "no",
          "label": "No"
        },
        {
          "value": "ungradable",
          "label": "Ungradable"
        }
      ]
    },
    {
      "label": "Contour complete - SRF?",
      "key": "pres_SRF_contour",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_path_SRF"
                },
                "yes"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_path_SRF"
                },
                "possible"
              ]
            }
          ]
        }
      },
```

```
      "required": true,
      "type": "radio",
      "values": [
        {
          "value": "yes",
          "label": "Yes"
        },
        {
          "value": "no",
          "label": "No"
        }
      ]
    },
    {
      "label": "Is SHRM present?",
      "key": "pres_path_SHRM",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_SRF_contour"
                },
                "yes"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_path_SRF"
                },
                "no"
              ]
            },
            {
```

```
            "==": [
              {
                "var": "pres_path_SRF"
              },
              "ungradable"
            ]
          }
        ]
      }
    },
    "required": true,
    "type": "radio",
    "values": [
      {
        "value": "yes",
        "label": "Yes",
        "requireMeasurements": [
          "SHRM"
        ],
        "measurementTools": [
          "FreehandRoi"
        ]
      },
      {
        "value": "possible",
        "label": "Possible",
        "requireMeasurements": [
          "SHRM"
        ],
        "measurementTools": [
          "FreehandRoi"
        ]
      },
      {
        "value": "no",
        "label": "No"
```

```
        },
        {
          "value": "ungradable",
          "label": "Ungradable"
        }
      ]
    },
    {
      "label": "Contour complete - SHRM?",
      "key": "pres_SHRM_contour",
      "conditional": {
        "json": {
          "or": [
            {
              "==": [
                {
                  "var": "pres_path_SRF"
                },
                "yes"
              ]
            },
            {
              "==": [
                {
                  "var": "pres_path_SRF"
                },
                "possible"
              ]
            }
          ]
        }
      },
      "required": true,
      "type": "radio",
      "values": [
        {
```

```
        "value": "yes",
        "label": "Yes"
      },
      {
        "value": "no",
        "label": "No"
      }
    ]
  },
  {
    "label": "Is PED present?",
    "key": "pres_path_PED",
    "conditional": {
      "json": {
        "or": [
          {
            "==": [
              {
                "var": "pres_path_SHRM"
              },
              "yes"
            ]
          },
          {
            "==": [
              {
                "var": "pres_path_SHRM"
              },
              "possible"
            ]
          },
          {
            "==": [
              {
                "var": "pres_SHRM_contour"
              },
```

```
              "yes"
            ]
          }
        ]
      }
    },
    "required": true,
    "type": "radio",
    "values": [
      {
        "value": "yes",
        "label": "Yes"
      },
      {
        "value": "possible",
        "label": "Possible"
      },
      {
        "value": "no",
        "label": "No"
      },
      {
        "value": "ungradable",
        "label": "Ungradable"
      }
    ]
  },
  {
    "label": "Which PED type?",
    "key": "pres_path_PED-type",
    "conditional": {
      "json": {
        "==": [
          {
            "var": "pres_path_PED"
          },
```

```
            "yes"
          ]
        }
      },
      "required": true,
      "type": "radio",
      "values": [
        {
          "value": "Fibrovascular",
          "label": "Fibrovascular (F)",
          "requireMeasurements": [
            "PED-F"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
        },
        {
          "value": "serous",
          "label": "Serous (S)",
          "requireMeasurements": [
            "PED-S"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
        },
        {
          "value": "drusenoid",
          "label": "Drusenoid (D)",
          "requireMeasurements": [
            "PED-D"
          ],
          "measurementTools": [
            "FreehandRoi"
          ]
```

```
          }
        ]
      },
      {
        "label": "Additional Notes",
        "key": "notes",
        "conditional": {
          "json": {
            "==": [
              {
                "var": "interpretable"
              },
              "yes"
            ]
          }
        },
        "required": false,
        "type": "text"
      }
    ]
  }
}
```

Download PDF