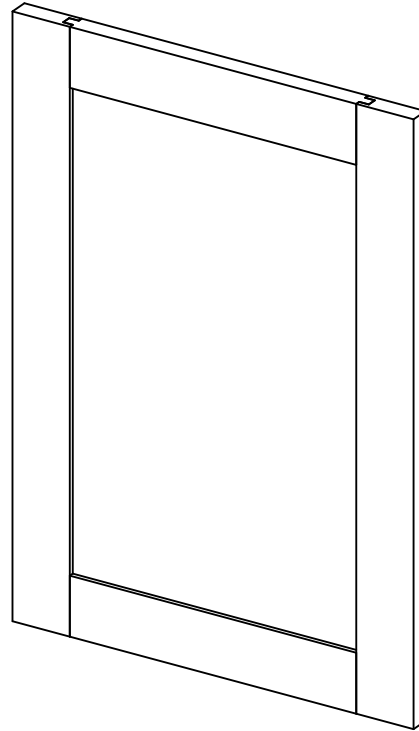


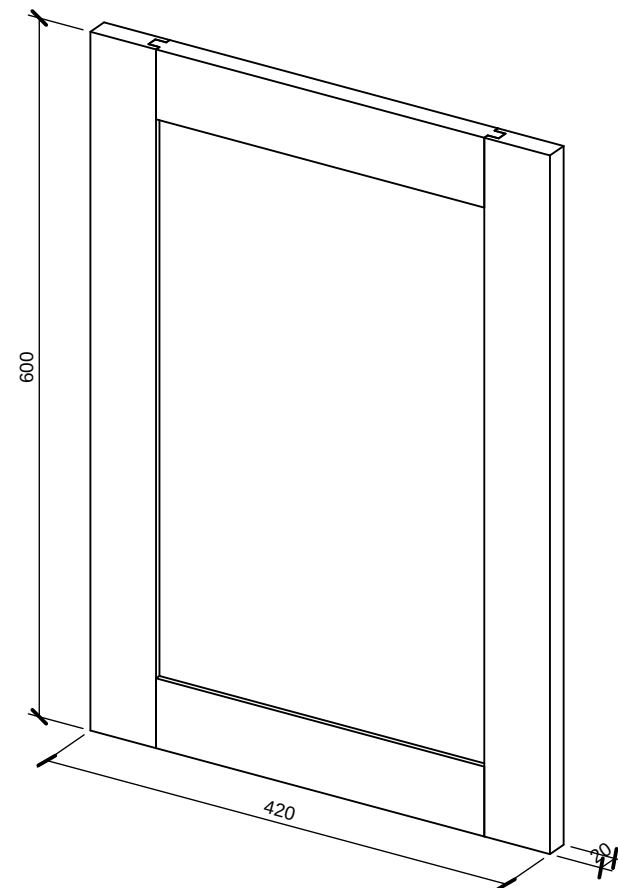
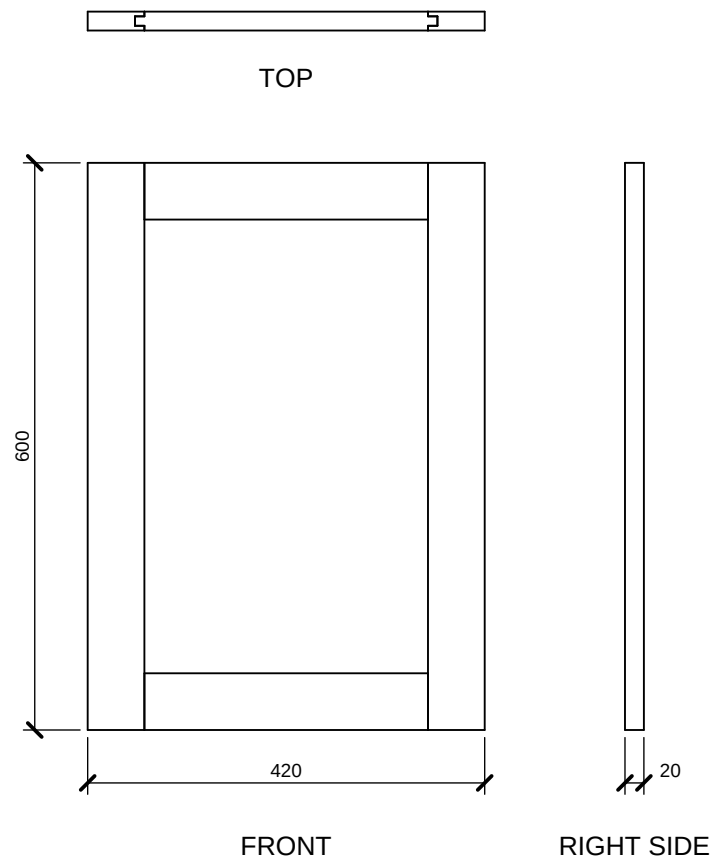
Frame and panel



parts	5 (3 different)
overall	420 x 20 x 600 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

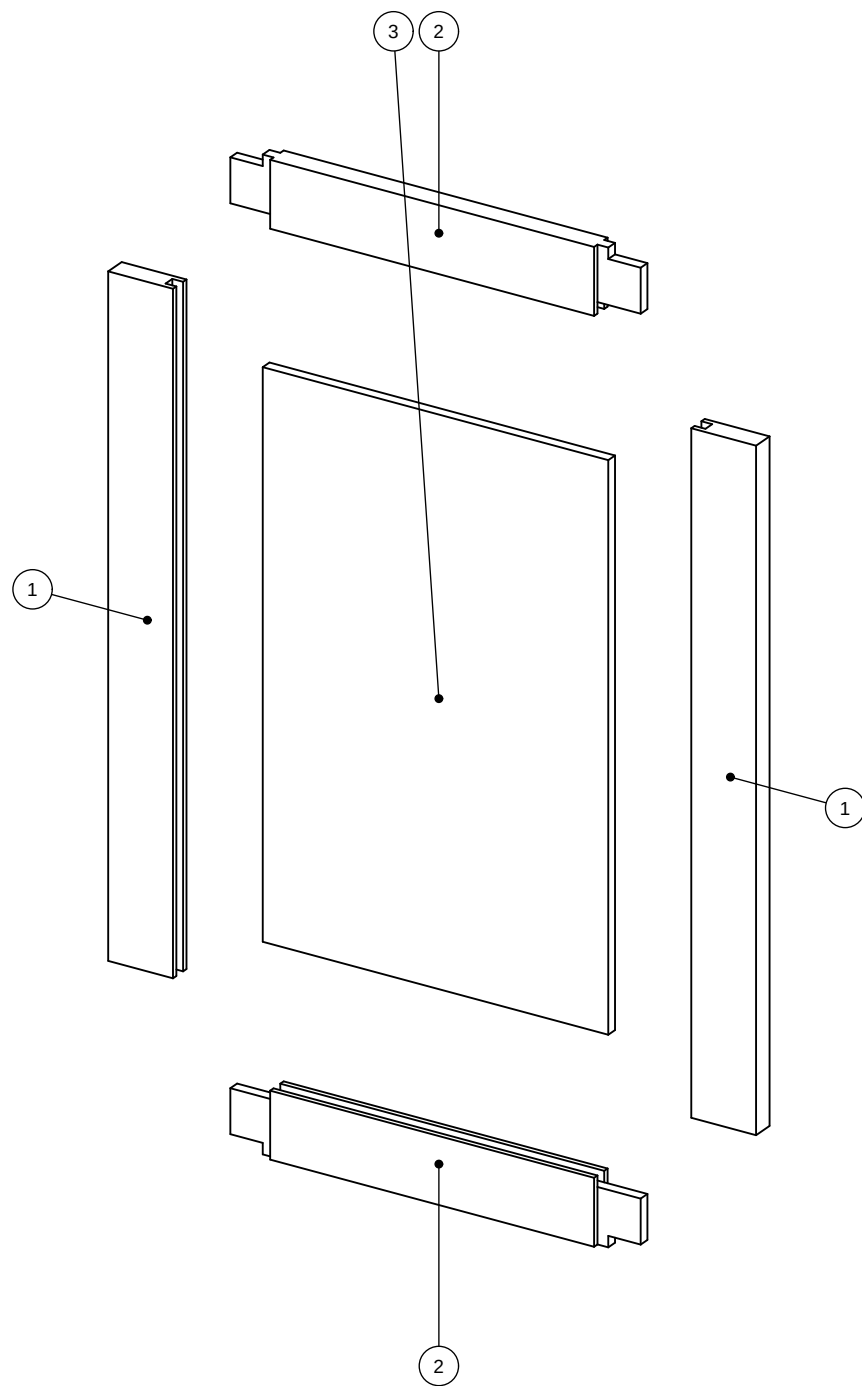
Contents

Drawings	3
Assembly	3
Parts	4
1 stile	5
2 rail	6
3 panel	7
 Cut list	 8
 Appendix: source	 9



ISOMETRIC

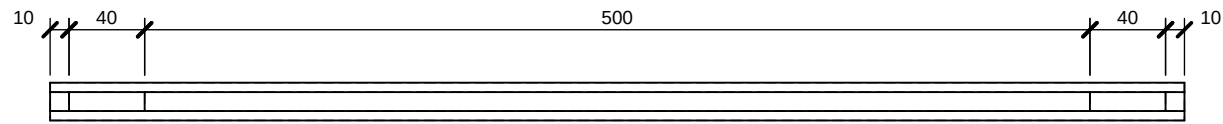
assembly	frame_and_panel
parts	5 (3 different)
overall	420 x 20 x 600 (W x D x H)
material	oak
scale	1:8, mm



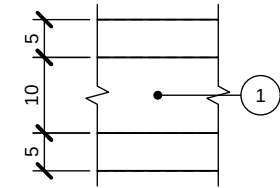
EXPLODED

no.	part	qty	cut size (L x W x T)	material
1	stile	2	600 x 60 x 20	oak
2	rail	2	380 x 60 x 20	oak
3	panel	1	500 x 320 x 10	oak

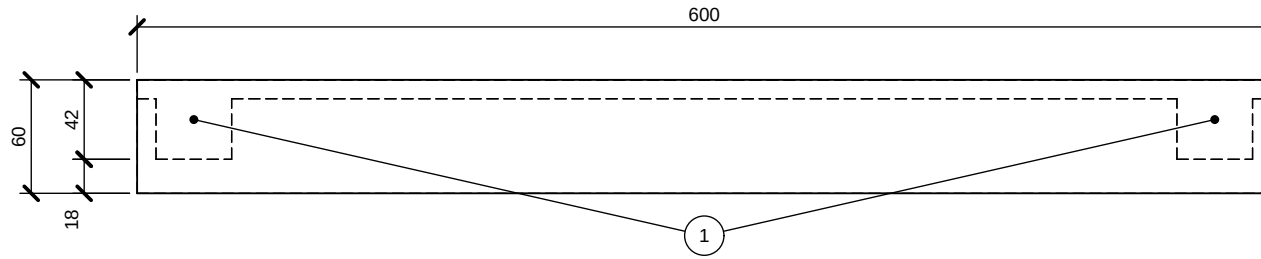
assembly	frame_and_panel
parts	5 (3 different)
overall	420 x 20 x 600 (W x D x H)
material	oak
scale	not to scale, mm



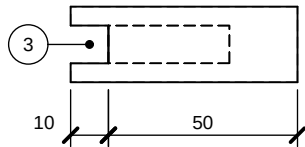
BACK EDGE



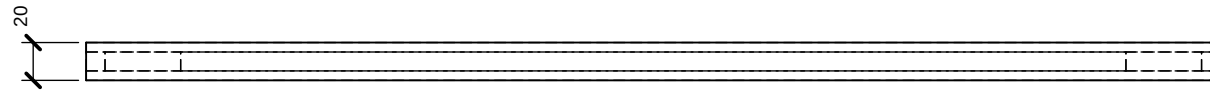
BACK EDGE: 1-4 1:1



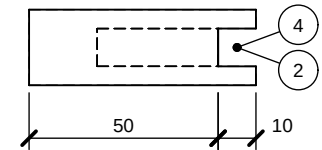
TOP FACE



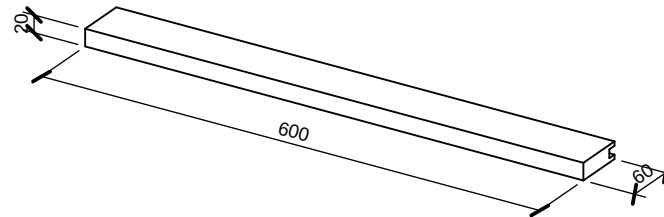
LEFT END 1:2



FRONT EDGE



RIGHT END 1:2



ISOMETRIC

2 pieces: stile_left, stile_right

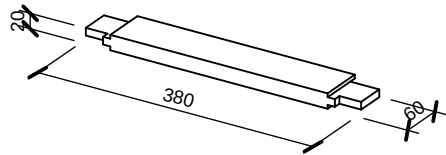
1 2 x mortise 40 x 10 mm, 42 deep, in back_edge

2 haunch mortise 10 x 10 mm, 10 deep, in back_edge at right_end

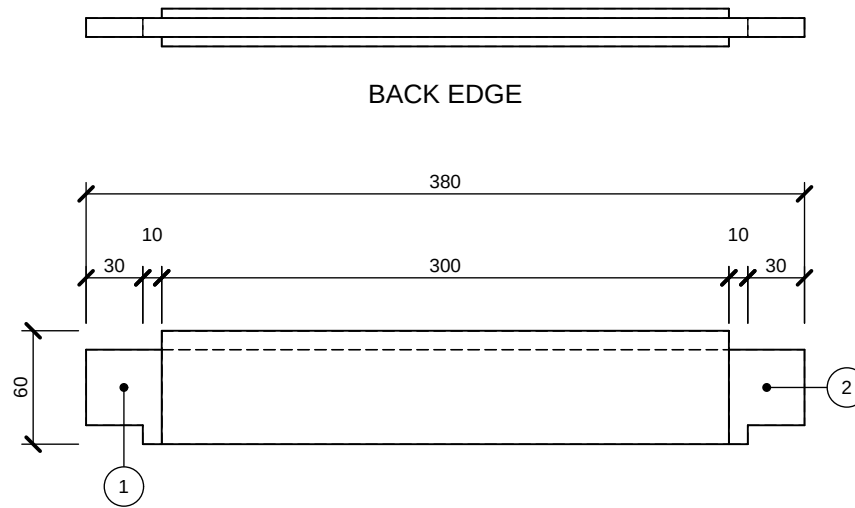
3 haunch mortise 10 x 10 mm, 10 deep, in back_edge at left_end

4 groove 10 mm, 10 deep, in back_edge

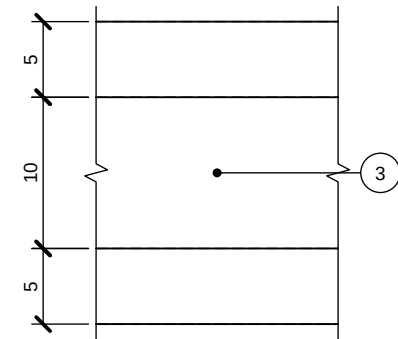
part	stile
of	frame_and_panel
make	2
cut size	600 x 60 x 20
material	oak
scale	1:4, mm



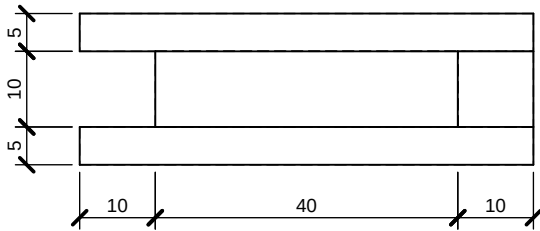
ISOMETRIC



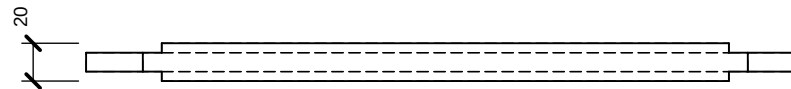
TOP FACE



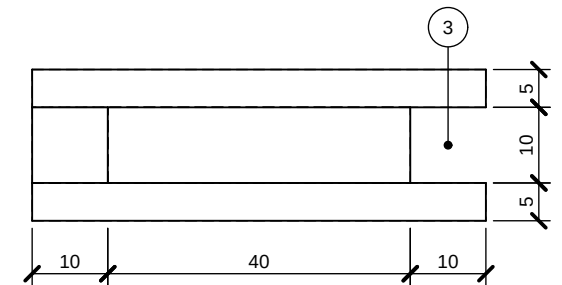
BACK EDGE: 3 2:1



LEFT END 1:1



FRONT EDGE



RIGHT END 1:1

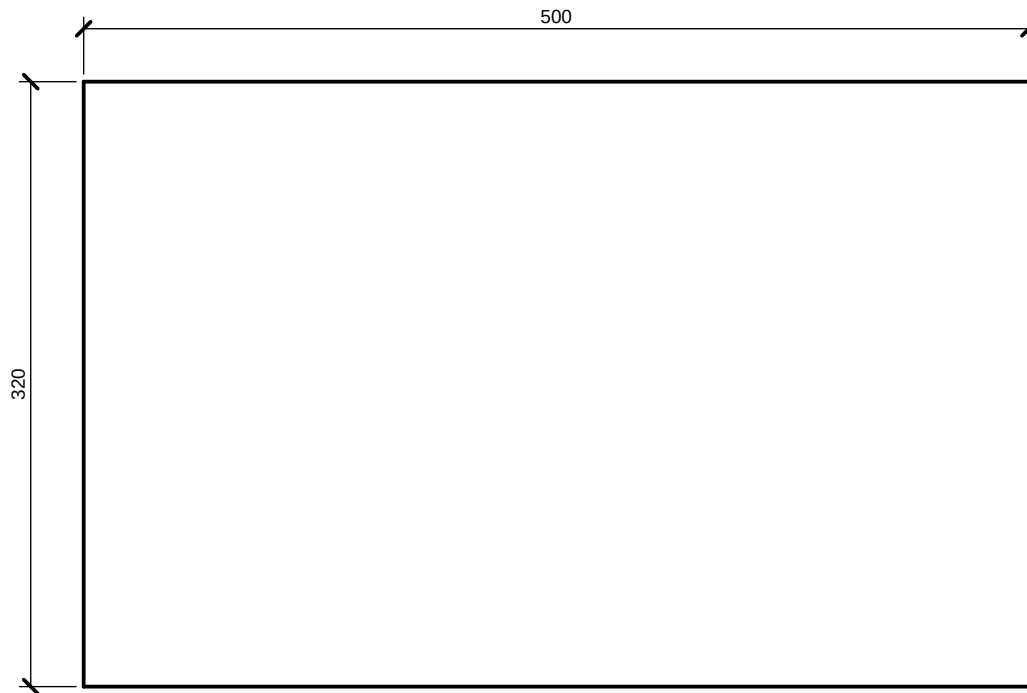
2 pieces: rail_top, rail_bottom

1 haunched tenon 40 x 10 mm, 40 long, haunch 10 long, on left_end

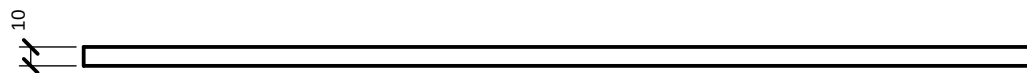
2 haunched tenon 40 x 10 mm, 40 long, haunch 10 long, on right_end

3 groove 10 mm, 10 deep, in back_edge

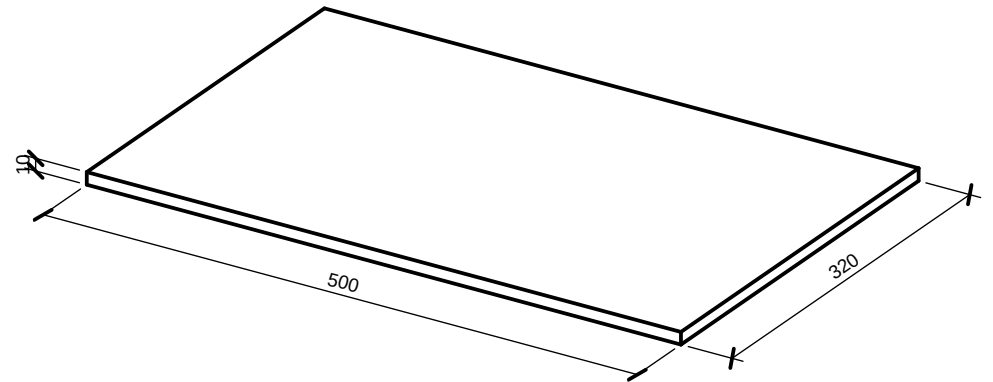
part	rail
of	frame_and_panel
make	2
cut size	380 x 60 x 20
material	oak
scale	1:4, mm



TOP FACE



FRONT EDGE



ISOMETRIC

part	panel
of	frame_and_panel
make	1
cut size	500 x 320 x 10
material	oak
scale	1:4, mm

Cut list

```
stile  600 x 60 x 20 mm  oak, 2 pieces: stile_left, stile_right
- 2 x mortise 40 x 10 mm, 42 deep, in back_edge
- haunch mortise 10 x 10 mm, 10 deep, in back_edge at right_end
- haunch mortise 10 x 10 mm, 10 deep, in back_edge at left_end
- groove 10 mm, 10 deep, in back_edge
rail   380 x 60 x 20 mm  oak, 2 pieces: rail_top, rail_bottom
- haunched tenon 40 x 10 mm, 40 long, haunch 10 long, on left_end
- haunched tenon 40 x 10 mm, 40 long, haunch 10 long, on right_end
- groove 10 mm, 10 deep, in back_edge
panel  500 x 320 x 10 mm  oak
```

Appendix: source

Made with woodwright 0.9.0, build123d 0.13.0, Python 3.14.4.

To make these plans again:

```
uvx --python 3.14 woodwright==0.9.0 export frames.frame_and_panel
```

frames.frame_and_panel, in woodwright/examples/frames.py

```
def frame_and_panel(
    stile_length: float = 600,
    rail_length: float = 300,
    width: float = 60,
    shoulder: float = 10,
    stock: Stock = OAK,
    panel_stock: Stock = THIN_OAK,
) -> Assembly:
    """A mortise-and-tenon frame with a panel in grooves along its inside edges.

    The panel's design size is the opening of the frame; the grooves add to it.
    The grooves run through the stiles, so the tenons are as thick as the
    grooves are wide and have haunches filling the grooves out to the ends.
    """

    def corner(rail_end: SideRef, stile_edge: SideRef, position: Position, outer: SideRef) -> Joint:
        return MortiseTenon(
            tenon=rail_end,
            mortise=stile_edge,
            position=position,
            shoulder=shoulder,
            tenon_thickness=panel_stock.thickness,
            haunch=outer,
        )

    asm, (stile_left, stile_right, rail_top, rail_bottom) = _frame(
        "frame_and_panel", corner, stile_length, rail_length, width, stock
    )
    panel = asm.add(Board("panel", panel_stock, length=stile_length - 2 * width, width=rail_length))
    for groove, edge in [
```

```
(stile_left.back_edge, panel.front_edge),  
(stile_right.front_edge, panel.back_edge),  
(rail_top.back_edge, panel.right_end),  
(rail_bottom.front_edge, panel.left_end),  
]:  
    asm.join(GrooveJoint(groove=groove, edge=edge))  
return asm
```