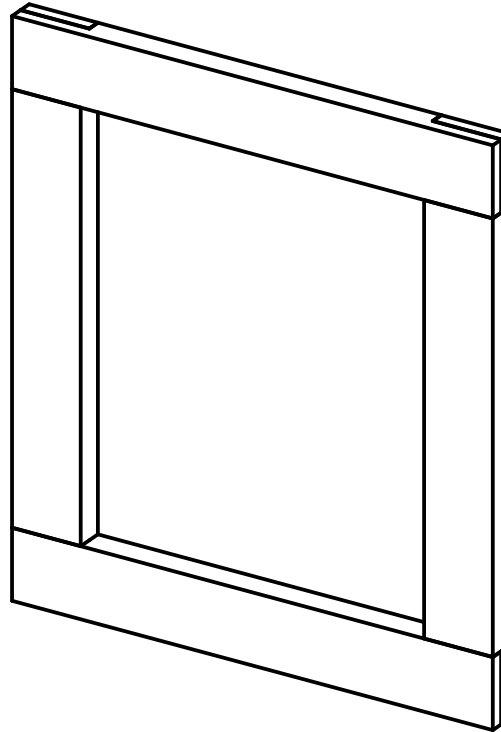


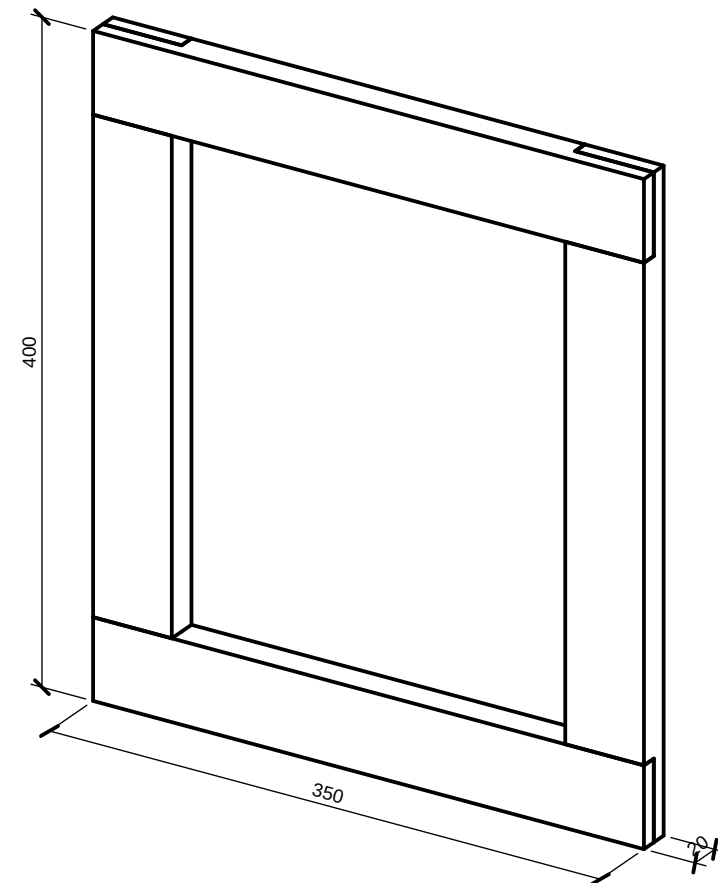
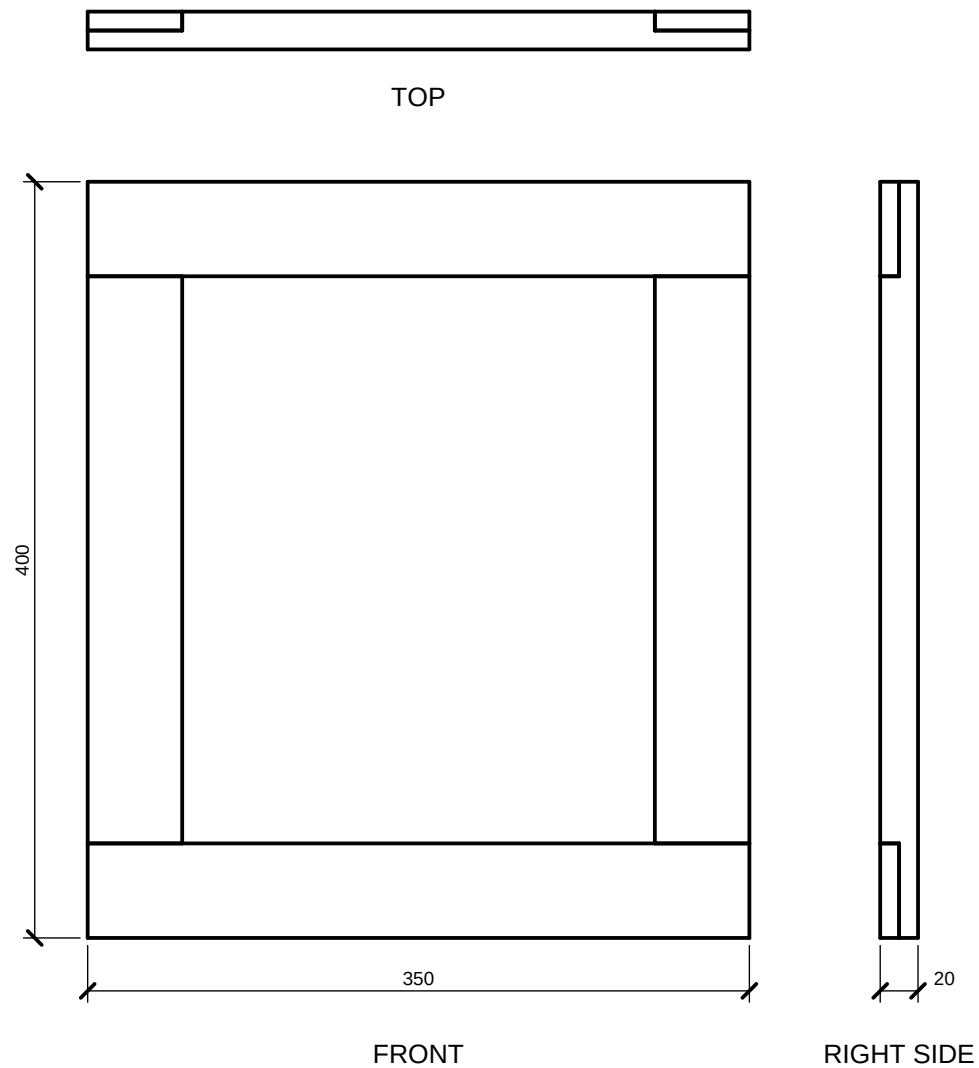
Half lap frame



parts	4 (2 different)
overall	350 x 20 x 400 (W x D x H)
material	oak
date	2026-10-25
made with	woodwright 0.9.0

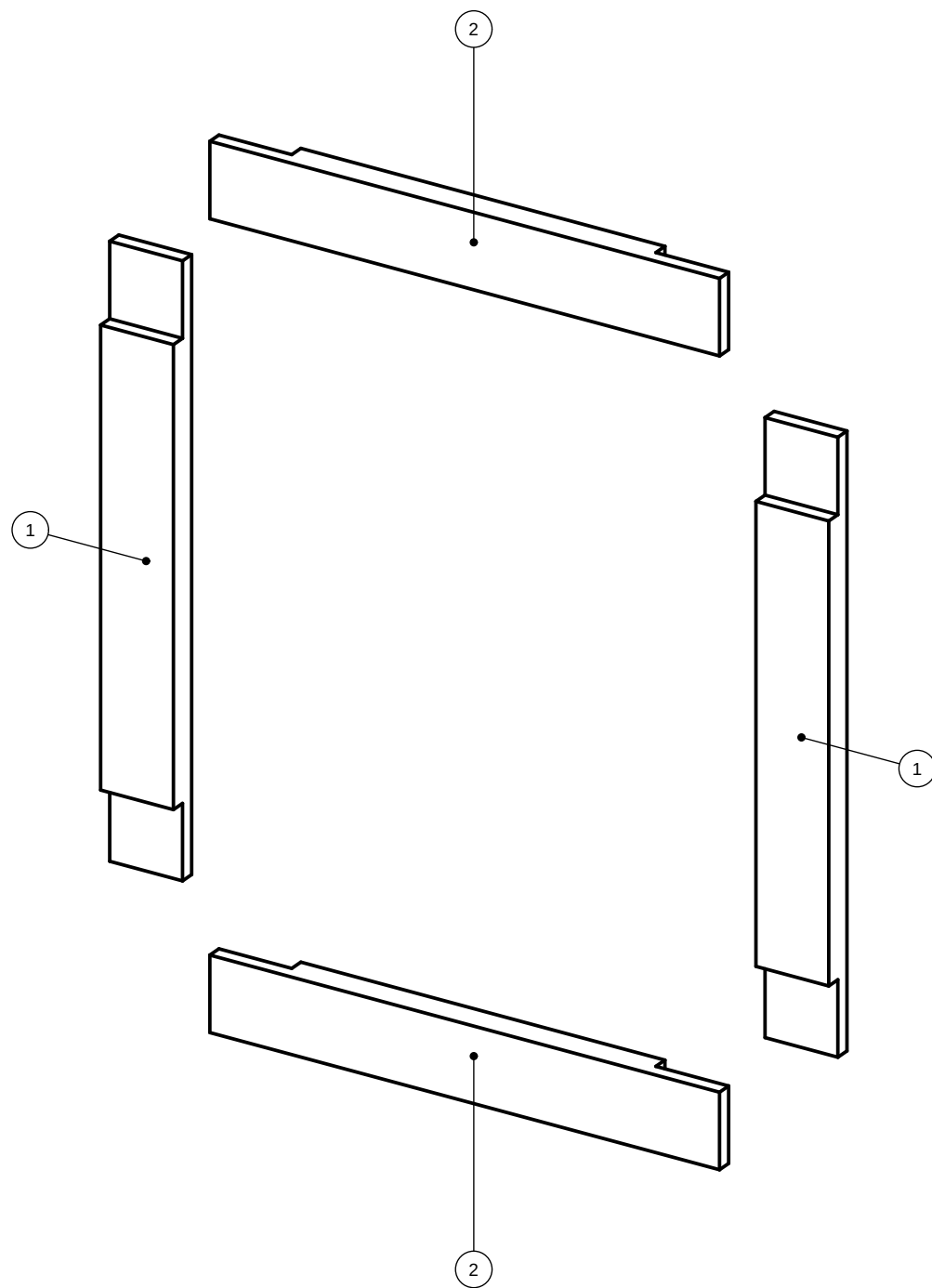
Contents

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



ISOMETRIC

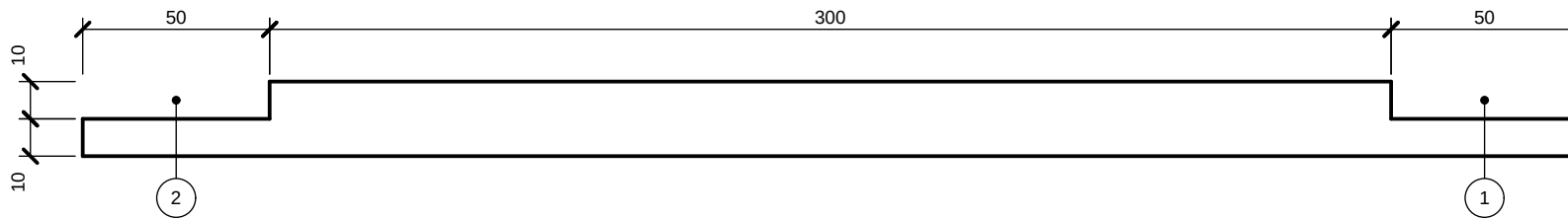
assembly	half_lap_frame
parts	4 (2 different)
overall	350 x 20 x 400 (W x D x H)
material	oak
scale	1:4, mm



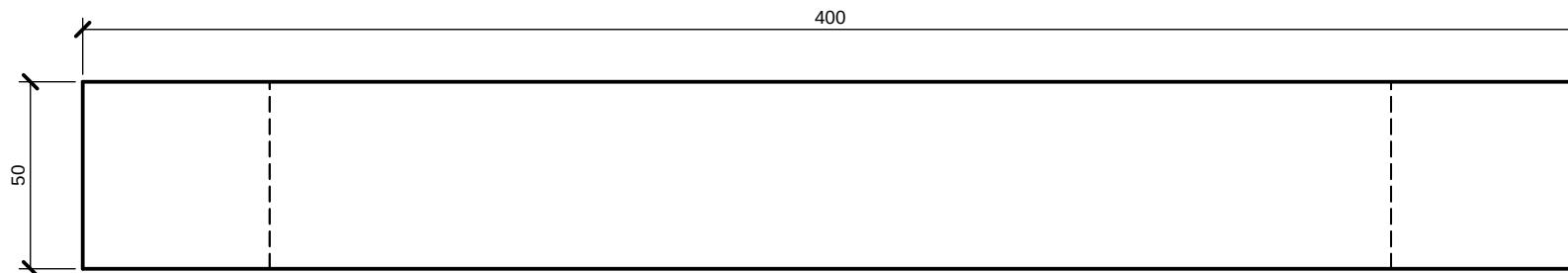
EXPLODED

no.	part	qty	cut size (L x W x T)	material
1	stile	2	400 x 50 x 20	oak
2	rail	2	350 x 50 x 20	oak

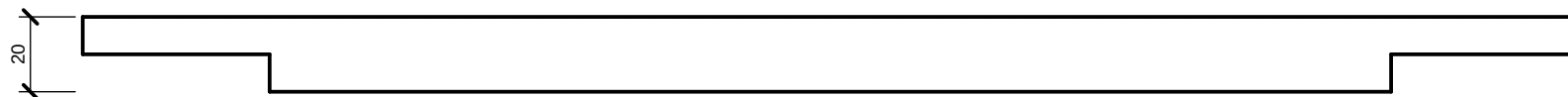
assembly	half_lap_frame
parts	4 (2 different)
overall	350 x 20 x 400 (W x D x H)
material	oak
scale	not to scale, mm



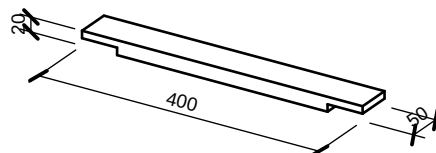
BACK EDGE



TOP FACE



FRONT EDGE



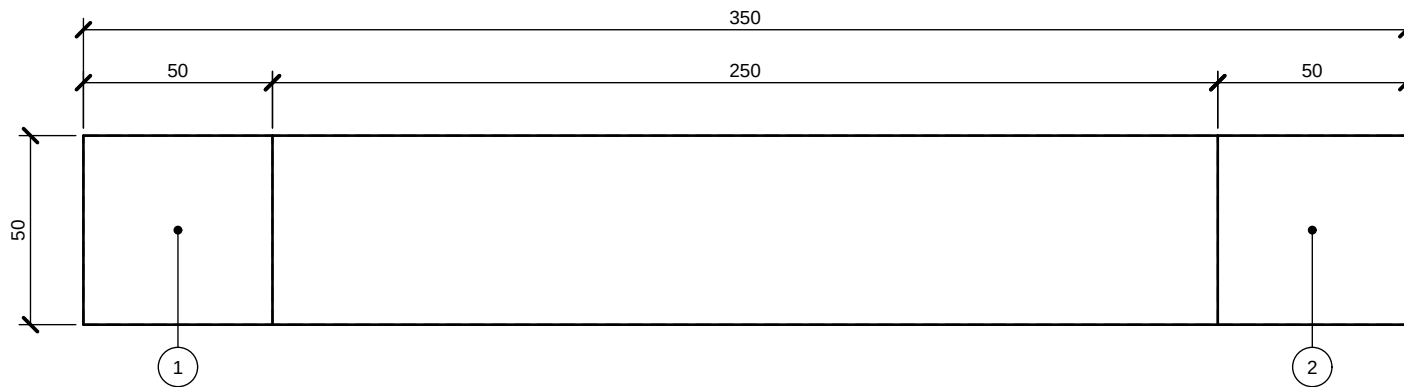
ISOMETRIC

2 pieces: stile_left, stile_right

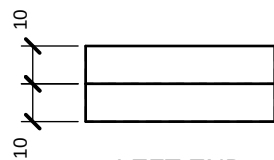
1 end rabbet 50 mm, 10 deep, in bottom_face at right_end

2 end rabbet 50 mm, 10 deep, in bottom_face at left_end

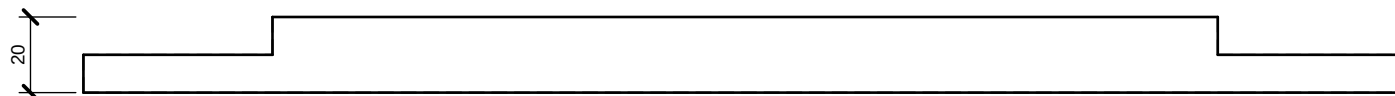
part	stile
of	half_lap_frame
make	2
cut size	400 x 50 x 20
material	oak
scale	1:2, mm



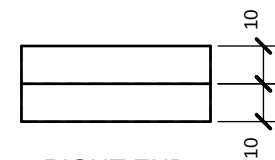
TOP FACE



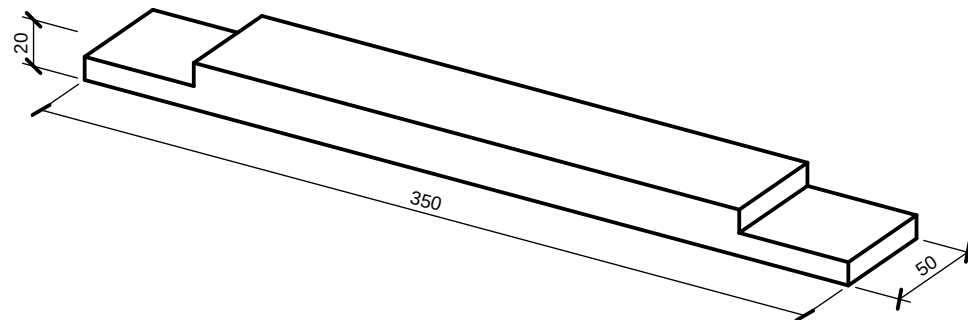
LEFT END



FRONT EDGE



RIGHT END



ISOMETRIC

- 2 pieces: rail_top, rail_bottom
 1 lap 10 mm thick, 50 long, on left_end
 2 lap 10 mm thick, 50 long, on right_end

part	rail
of	half_lap_frame
make	2
cut size	350 x 50 x 20
material	oak
scale	1:2, mm

Cut list

stile 400 x 50 x 20 mm oak, 2 pieces: stile_left, stile_right
- end rabbet 50 mm, 10 deep, in bottom_face at right_end
- end rabbet 50 mm, 10 deep, in bottom_face at left_end
rail 350 x 50 x 20 mm oak, 2 pieces: rail_top, rail_bottom
- lap 10 mm thick, 50 long, on left_end
- lap 10 mm thick, 50 long, on right_end

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.half_lap_frame
```

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

```
def half_lap_frame(
    stile_length: float = 400,
    rail_length: float = 250,
    width: float = 50,
    stock: Stock = OAK,
) -> Assembly:
    """A frame with half-lap corners."""

    def corner(rail_end: SideRef, stile_edge: SideRef, position: Position, outer: SideRef) -> Joint:
        return HalfLap(lap=rail_end, socket=stile_edge, position=position)

    asm, _ = _frame("half_lap_frame", corner, stile_length, rail_length, width, stock)
    return asm
```