

# LUMEN: Carotid Exam Engine — Full Lifecycle Overview (Phase 1A + 1B)

## From Template to Final Report

## CAROTID WORKFLOW OVERVIEW

### 1. JSON TEMPLATE LOADING

reports/exam\_templates/carotid.json (structure)



get\_template(exam\_type="carotid", site="mountsinai")



returns JSON describing all carotid segments

e.g. {"segments": [{ "id": "prox\_ica\_right", ... }, ...]}

### 2. EXAM + SEGMENT OBJECTS FROM TEMPLATE

create\_exam\_from\_template(exam\_type, site, patient\_data, create



Creates Exam object → with Segment and Measurement children

- Each Segment: name, side, vessel
- Each Measurement: initialized with PSV, EDV, CCA\_PSV, etc. as bl

### 3. SITE-LEVEL LOGIC LOADS CRITERIA

load\_carotid\_criteria(site="mountsinai")



reads → reports/site/mountsinai/criteria/carotid.json

Returns rules like:

```
{
  "stenosis_thresholds": { "0_19": { "psv_max": 104 }, ... },
  "vertebral_rules": { "steal_direction": "retrograde", ... }
}
```

### 4. SEGMENTS INTO DICTIONARY FORM

```

for segment in exam.segments:
    segment_dict[segment.name] = {
        "psv": segment.measurement.psv,
        "edv": segment.measurement.edv,
        ...
    }

```

These are typed using:

→ ArterialSegmentBase

→ CarotidSegmentDict (adds ica\_cca\_ratio, stenosis\_category, etc.)

## 5. CALCULATOR LOGIC APPLIED

```

calculator = CarotidCalculator(segment_dict, carotid_criteria)
calculator.run_all()

```

► For each segment:

- compute\_ica\_cca\_ratio() → adds "ica\_cca\_ratio"
- apply\_stenosis\_logic() → adds "stenosis\_category" + "stenosis\_no"
- interpret\_vertebral\_waveform() → adds "vertebral\_comment"

## 6. OUTPUT + INTEGRATION

```

calculator.get_segment_data()
    → gives structured dict[str, CarotidSegmentDict] with all fields

```

```

calculator.export_json()
    → gives pretty-printed JSON (debug/log use)

```

↓

Optional:

generate\_conclusion(exam) → builds editable findings summary like:

"Prox ICA R: Findings consistent with 60–79% stenosis."

"Left Vertebral: Retrograde vertebral flow consistent with subclavian

↓

Final Output:

- Frontend Form (Formik + MeasurementTable)
- PDF (WeasyPrint via PDF template)
- HL7 Payload (NextGen via Mirth)

File / Component	Purpose
`carotid.json`	Defines the structure of all carotid segments
`get_template()`	Loads template based on exam_type + site
`create_exam_from_template()`	Creates DB models from that template
`load_carotid_criteria()`	Loads threshold rules per site
`ArterialSegmentBase`	Shared segment fields (psv, edv, waveform, etc.)
`CarotidSegmentDict`	Extended carotid-specific fields
`CarotidCalculator`	Executes stenosis, ICA/CCA, vertebral logic