

1 main — MIR Walkthrough

Purpose: TODO: Describe why this walkthrough exists

1.1 Source Context

```
    assert!(sum() == 42);  
}
```

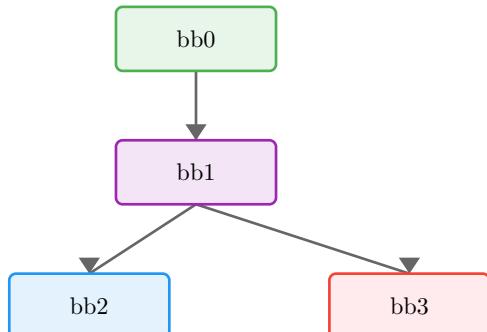
1.2 Function Overview

- **Function:** main
- **Basic blocks:** 4
- **Return type:** ()
- **Notable properties:**
 - Contains panic path
 - Introduces borrows
 - Has conditional branches

1.3 Locals

Local	Type	Notes
0	()	Return place
1	{closure@tests/integration/programs/closure-no-args.rs:2:15: 2:24}	
2	u32	
3	&{closure@tests/integration/programs/closure-no-args.rs:2:15: 2:24}	
4	!	

1.4 Control-Flow Overview



1.5 Basic Blocks

1.5.1 bb0 — entry

Entry point of the function.

MIR	Annotation
_3 = &_1	Shared borrow
→ _2 = call(move _3, ()) → bb1	Call call

1.5.2 bb1 — branch point

MIR	Annotation

→ switch(move _2) [42→bb2; else→bb3]	Branch on move _2
--------------------------------------	-------------------

1.5.3 bb2 — return / success

Normal return path.

MIR	Annotation
→ return	Return from function

1.5.4 bb3 — panic path

Panic/diverging path.

MIR	Annotation
→ _4 = panic([16 bytes])	Call panic

1.6 Key Observations

TODO: Add bullet points summarizing what this MIR teaches

-
-

1.7 Takeaways

TODO: One or two sentences to generalize this example

2 {closure#0} — MIR Walkthrough

Purpose: TODO: Describe why this walkthrough exists

2.1 Source Context

```
fn main() {
    let sum = || -> u32 { 42 };
```

2.2 Function Overview

- **Function:** {closure#0}
- **Basic blocks:** 1
- **Return type:** u32

2.3 Locals

Local	Type	Notes
0	u32	Return place
1	&{closure@tests/integration/programs/closure-no-args.rs:2:15: 2:24}	

2.4 Control-Flow Overview

```
bb0
```

2.5 Basic Blocks

2.5.1 bb0 — entry

Entry point of the function.

MIR	Annotation
_0 = 42	Load constant
→ return	Return from function

2.6 Key Observations

TODO: Add bullet points summarizing what this MIR teaches

-
-

2.7 Takeaways

TODO: One or two sentences to generalize this example

