

1 main — MIR Walkthrough

Purpose: TODO: Describe why this walkthrough exists

1.1 Source Context

```
fn main() {
    let tup:(i32, i32) = (42, 99);

    assert!(tup == (42, 99));
}
```

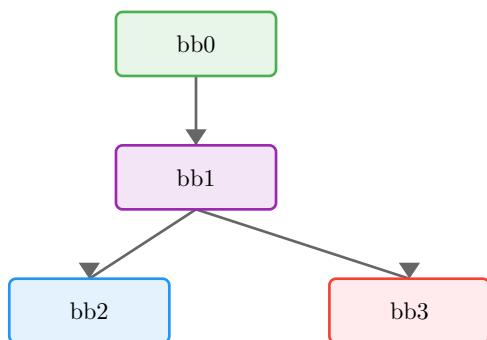
1.2 Function Overview

- **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	(i32, i32)	
2	bool	
3	&(i32, i32)	
4	&(i32, i32)	
5	!	

1.4 Control-Flow Overview



1.5 Basic Blocks

1.5.1 bb0 — entry

Entry point of the function.

MIR	Annotation
_1 = Tuple(42, 99)	Construct aggregate
_3 = &_1	Shared borrow
_4 = 0	Load constant

$\rightarrow _2 = \text{eq}(\text{move } _3, \text{ move } _4) \rightarrow \text{bb1}$	Call eq
---	---------

1.5.2 bb1 — branch point

MIR	Annotation
$\rightarrow \text{switch}(\text{move } _2) \backslash [0 \rightarrow \text{bb3}; \text{ else} \rightarrow \text{bb2} \backslash]$	Branch on move $_2$

1.5.3 bb2 — return / success

Normal return path.

MIR	Annotation
$\rightarrow \text{return}$	Return from function

1.5.4 bb3 — panic path

Panic/diverging path.

MIR	Annotation
$\rightarrow _5 = \text{panic}(\backslash [16 \text{ bytes}\backslash])$	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

