

# 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:** () (0 bytes, align 1)
- **Notable properties:**
  - Contains panic path
  - Introduces borrows
  - Has conditional branches

## 1.3 Locals

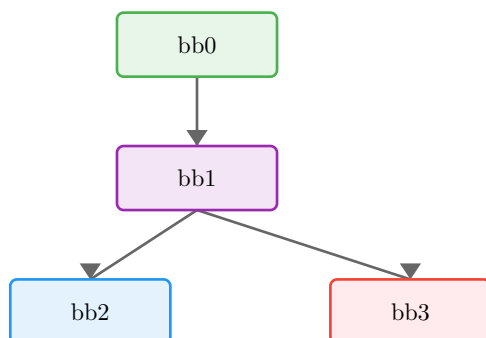
| Local | Type   | Notes        |
|-------|--|--------------|
| 0     | () (0 bytes, align 1)  | Return place |
| 1     | {closure@tests/integration/programs/closure-no-args.rs:2:15: 2:24}                     |              |
| 2     | Uint(U32)  |              |
| 3     | &{closure@tests/integration/programs/closure-no-args.rs:2:15: 2:24} (8 bytes, align 8) |              |
| 4     | ()   |              |

## 1.4 Borrows

| # | Borrow | Kind | Created At | Borrowed |
|---|--------|------|------------|----------|
| 0 | _3     | &    | bb0[0]     | _1       |

*Borrows are tracked conservatively: active from creation until reassignment or scope end.*

## 1.5 Control-Flow Overview



## 1.6 Basic Blocks

### 1.6.1 bb0 — entry

*Entry point of the function.*

| MIR   | Annotation    |
|---|---------------|
| <code>_3 = &amp;_1</code>                   | Shared borrow |
| <code>→ _2 = call(move _3, ()) → bb1</code> | Call call     |

#### 1.6.2 bb1 — branch point

| MIR   | Annotation        |
|---|-------------------|
| <code>→ switch(move _2) [42→bb2; else→bb3]</code> | Branch on move _2 |

#### 1.6.3 bb2 — return / success

*Normal return path.*

| MIR                   | Annotation           |
|-----------------------|----------------------|
| <code>→ return</code> | Return from function |

#### 1.6.4 bb3 — panic path

*Panic/diverging path.*

| MIR                                   | Annotation |
|---------------------------------------|------------|
| <code>→ _4 = panic([16 bytes])</code> | Call panic |

### 1.7 Key Observations

TODO: Add bullet points summarizing what this MIR teaches

- 
- 

### 1.8 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:** Uint(U32)

### 2.3 Locals

| Local | Type   | Notes        |
|-------|--|--------------|
| 0     | Uint(U32)  | Return place |
| 1     | &{closure@tests/integration/programs/closure-no-args.rs:2:15: 2:24} (8 bytes, align 8) |              |

### 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

