

# Inputs and Outputs

EGL: A Domain-Specific Language for Specifying and Enforcing  
Ethical Requirements in Large Language Models

## Example 1

### Policy Definition and Input:

```
@debug
policy my_policy(post):
  redact: email "*"
end
```

### Input:

Write to author@university.edu

### Basic Output:

Write to \*\*\*\*\*  
[DEBUG] redact: email (1)

### JSON Output:

```
{
  "mode": "post",
  "status": "OK",
  "return_code": 0,
  "transformed_text": "Write to *****",
  "actions_applied": [
    {"action": "redact", "evaluated": "email", "count": 1}
  ],
  "totals": {"forbid": 0, "redact": 1, "append": 0},
  "debug": {"enabled": true}
}
```

### Example 2

#### Policy Definition and Input:

```
@debug
policy phone_block(pre):
  forbid: phone_number
end
```

#### Input:

My phone is 5555-9213.

#### Basic Output:

FORBIDDEN OUTPUT  
[DEBUG] forbid: phone\_number (1)

#### JSON Output:

```
{
  "mode": "pre",
  "status": "FORBIDDEN",
  "return_code": -2,
  "transformed_text": "My phone is 5555-9213.",
  "actions_applied": [
    {"action": "forbid", "evaluated": "phone_number", "count": 1}
  ],
  "totals": {"forbid": 1, "redact": 0, "append": 0},
  "debug": {"enabled": true}
}
```

### Example 3

#### Policy Definition and Input:

```
@debug
policy redact_email(post):
  append: email "There is an email on this message."
end
```

#### Input

Please contact alice@company.org for details.

#### Basic Output

Please contact alice@company.org for details.  
There is an email on this message.  
[DEBUG] append: email (1)

### Structured JSON Output

```
{
  "mode": "post",
  "status": "OK",
  "return_code": 0,
  "transformed_text": "Please contact alice@company.org
for details. There is an email on this message.",
  "actions_applied": [
    {"action": "append", "evaluated": "email", "count": 1}
  ],
  "totals": {"forbid": 0, "redact": 0, "append": 1}
}
```