


# Actions

---

 [ulfurinn.github.io/wongi-engine/03-actions.html](https://ulfurinn.github.io/wongi-engine/03-actions.html)

## gen

---

gen x, y, z

Asserts a fact according to the provided template and links it to the current token. If all linked tokens are invalidated and the same fact has not been asserted manually, it is retracted.

## action

---

```
action { |token|  
  # ...  
}
```

```
action activate: ->(token) { }, deactivate: ->(token) { }
```

```
action ActionClass, *args, &block  
class ActionClass  
  def initialize(*args, &block)  
  def execute(token)  
    # ...  
  end  
  def deexecute(token)  
    # ...  
  end  
end
```

A piece of code executed upon activating and/or deactivating the production node.

The action class may define the following accessors that will be populated with relevant data:

- **rete** : the engine instance
- **rule** : the rule in which the action is included
- **name** : the extension clause through which the action was instantiated
- **production** : the production node

A short way to define all 4 is to inherit from `Wongi::Engine::Action`, but this is not necessary.

## collect

---

collect collection\_name, variable

`engine.collection(collection_name)` will return a **uniq** ed array of all encountered values of the variable.

## error

---

```
error message
error { |bound_variables|
  # ...
  error_message
}
```

`engine.errors` will return all generated errors. Useful, for example, to detect contradictory facts.

## trace

---

trace options

Prints a message each time the production node is activated. The following options are recognized:

- `:values => false` : whether to print all bound variables
- `:io => $stdout` : where to write the message
- `:generation => false` : whether this rule's `gen` actions should print debug traces, too. Must precede the `gen` actions in this case.
- `:tracer` : a class that defines `#trace(x)` . The content of `x` will vary depending on the action.